

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Ambrožič

**Večkriterijska optimizacija pametnih
stavb z evolucijskim algoritmom**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Marko Robnik Šikonja

SOMENTOR: izr. prof. dr. Bogdan Filipič

Ljubljana 2015

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Pametne stavbe vsebujejo avtomatske sisteme za spremljanje in nadzor njihovega delovanja. Med seboj povezani nadzorni sistemi upravljajo z razsvetljavo, varnostnimi sistemi, temperaturo, nadzorujejo predvajanje multimedijskih vsebin, vhodna vrata, kroženje in tok zraka v prostorih, obnašanje gospodinjskih aparatov itd. Cilj nadzornih sistemov je uporabnikom zagotoviti čim več udobja, obenem pa tudi energetske učinkovitost in varnost. Ker so si cilji večkrat v nasprotju, je za optimizacijo delovanja kontrolnih sistemov smiselno poseči po večkriterijski optimizaciji. V nalogi predstavite problem upravljanja pametnih stavb in pristop k njihovem upravljanju z večkriterijsko optimizacijo s pomočjo evlucijskega algoritma. Predlagajte predstavitev problema in genetske operatorje za njegovo reševanje z evlucijskim algoritmom. Pristop demonstrirajte na testni postavitvi pametne stavbe in rezultate ovrednotite.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Nejc Ambrožič, z vpisno številko **63120220**, sem avtor diplomskega dela z naslovom:

Večkriterijska optimizacija pametnih stavb

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Marka Robnika Šikonje in somentorstvom izr. prof. dr. Bogdana Filipiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 12. septembra 2015

Podpis avtorja:

Zahvaljujem se somentorju izr. prof. dr. Bogdanu Filipiču za možnost sodelovanja z Institutom “Jožef Stefan” in mentorstvo na področju evolucijskih algoritmov ter pri razvoju algoritma.

Posebna zahvala gre mentorju izr. prof. dr. Marku Robniku Šikonji za ves vložen trud, hitre odzive in vse nasvete pri izdelavi diplomske naloge.

Hvala Viktoriji za vse pogovore, nasvete in debate o pisanju ter vso izkazano podporo.

Mojim staršem

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Problem optimizacije pametnih stavb	5
2.1	Teoretična osnova	5
2.2	Obravnavani problem	8
3	Uporabljene optimizacijske metode	11
3.1	Evolucijski algoritem	11
3.1.1	Kromosom	18
3.1.2	Genetski operatorji	19
3.2	Stavba	21
3.2.1	Izvedba ljudem prijaznih sprememb	23
3.2.2	Upravljanje z elektriko	24
3.3	Arhitektura sistema	28
3.4	Postopek	31
4	Analiza in vrednotenje rezultatov	35
4.1	Analiza	35
4.1.1	Znanje o omrežju	35
4.1.2	Analiza vpliva začetnega stanja akumulatorja	39
4.2	Ovrednotenje rezultatov	44

Povzetek

Energetsko učinkovito delovanje bivanjskih stavb je ključnega pomena pri globalnem zmanjšanju porabe električne energije, saj povpraševanje po energiji, skladno s tehnološkim razvojem in povečano dostopnostjo tehnologije, narašča. Za optimiziranje delovanja pametnih stavb je v diplomskem delu predstavljen in razvit evolucijski algoritem, ki upošteva kriterija energetske učinkovitosti in uporabnikovega udobja. Namesto ene rešitve je moč poiskati Pareto optimalno fronto, na kateri se nahajajo kompromisne rešitve po več kriterijih. S poganjanjem simulacij v različnih testnih scenarijih smo pokazali, da algoritem na podlagi začetnega urnika, ki ga nastavi uporabnik, najde kompromisne rešitve, ki žrtvujejo udobje, a so stroškovno učinkovitejše.

Ključne besede: pametna stavba, večkriterijska optimizacija, evolucijski algoritem, Pareto optimalnost.

Abstract

Energy efficiency of residential buildings is crucial to the global reduction of electricity consumption, as energy demand, in line with technological development and the increased availability of technology, is increasing. To optimize the functioning of smart buildings, an evolutionary algorithm that takes into account the criteria of energy efficiency and comfort of the user, is presented and developed. Instead of a single solution, it is possible to find a Pareto optimal front, consisting of trade-off solutions. By simulating different test scenarios we show that the algorithm finds trade-off solutions based on the initial schedule set by the user. The obtained solutions sacrifice comfort, but are more cost-effective.

Keywords: smart house, multicriteria optimization, evolutionary algorithm, Pareto optimality.

Poglavje 1

Uvod

V zadnjih letih se je globalna družbena skrb nad okoljskimi težavami, kot sta globalno segrevanje in uničevanje ozonskega plašča, močno povečala. S tehnološkim napredkom in povečano dostopnostjo tehnologije se je močno povečala potreba po električni energiji. Za doseg ciljev zmanjšanja izpustov toplogrednih plinov, zastavljenih z Zakonom o ratifikaciji Kjotskega potočila k Okvirni konvenciji Združenih narodov o spremembah podnebja [1], je energetika ključno področje. Prispevek energetike k izpustom toplogrednih plinov znaša namreč kar okoli 80% [2]. V prihodnosti bodo povečana energetska učinkovitost sistemov in zmanjšane energetske potrebe končnih uporabnikov zelo pomembna pri zmanjšanju obsega izpustov toplogrednih plinov v ozračje. Ob upoštevanju rasti prebivalstva, povečane potrebe po storitvah in udobju, skupaj s spoznanjem, da ljudje dandanes večino časa preživimo v stavbah, je trend rasti porabe električne energije praktično neizbežen. Zmanjšanje porabe energije tako v komercialnem kot v bivalnem sektorju je še posebej pomembno, saj se povpraševanje po energiji v obeh sektorjih, za razliko od industrijskega in transportnega, povečuje. Poraba bivalnega sektorja se giblje med 20% in 40% vse porabljene energije in je tako že presegla porabo industrijskega in transportnega sektorja [3]. V letu 2010 so bivalni objekti v Združenih državah Amerike porabili kar 41% vse električne energije [4]. Smernice razvoja in družbena ozaveščenost o problema-

tiki globalnega ogrevanja na podlagi izpustov ogljikovega dioksida narekujejo smotrnejšo in učinkovitejšo porabo energije.

Razvoj varčnih in inteligentnih sistemov za vodenje objektov je tako ključnega pomena za zmanjšanje porabe električne energije, predvsem pri sistemih, ki porabijo največ energije. V to kategorijo uvrščamo sisteme za gretje, prezračevanje in hlajenje (angl. heating, ventilation and air conditioning – HVAC). Sistemi HVAC delujejo v zaprtih prostorih, njihov cilj je zagotoviti temperaturno udobje in sprejemljivo kakovost zraka. Delovanje sistema temelji na načelih termodinamike, mehanike tekočin in prenosa toplote [5]. V sodobnem domu naprave teh sistemov porabijo kar 50% energije, ki jo dom potrebuje [3]. Preizkušeni so bili že številni pristopi k optimizaciji sistemov HVAC. Glavne probleme pri tem predstavljajo večdimenzionalna in nelinearna narava sistemov, mešanica diskretnih in zveznih parametrov ter poleg zakonskih še številne omejitve sistemskih komponent [6].

V zadnjih dveh desetletjih je potekal stalen razvoj optimizacijskih metod, ki poskušajo omenjene probleme rešiti. Te metode zajemajo analitične pristope [7], metode nelinearnega [8] in dinamičnega programiranja [9] ter ume tne nevronske mreže [10]. Omenjeni pristopi so lahko učinkovito uporabljeni v nekaterih optimizacijskih problemih. Vsak pristop ima namreč določene prednosti, a hkrati probleme in omejitve pri uporabi v realnih sistemih. Hitro razvijajoča se paradigma hevrističnih optimizacijskih metod kaže dobre rezultate v praksi. Mednje štejemo pristope, kot so simulirano ohlajanje, optimizacija z rojem delcev, kolonija mravelj, iskanje s tabuji in evolucijski algoritmi. Za evolucijske algoritme je bilo ugotovljeno, da se pri optimizaciji sistemov HVAC obnesejo bolje kot druge hevristične metode [11, 12].

Najpopularnejši in najpogosteje uporabljen tip evolucijskih algoritmov so genetski algoritmi. Koncept genetskih algoritmov so razvili Holland in sodelavci v šestdesetih in sedemdesetih letih prejšnjega stoletja [13]. V posne manju naravne selekcije sledijo principu preživetja najmočnejšega, kjer šibki posamezniki izumrejo, močni pa imajo večjo možnost reprodukcije in prenosa svojega dednega materiala na prihodnje generacije. Na dolgi rok postanejo

osebki, ki nosijo najboljšo genetsko kombinacijo v populaciji, dominantni in tako razširijo svoj dedni material. Včasih med procesom evolucije pride do naključnih mutacij, ki lahko prinesejo dodatne prednosti za preživetje, slabe spremembe pa se izločijo z naravno selekcijo. Dejstvo, da genetski algoritmi temeljijo na populaciji, jih naredi primerne za reševanje večkriterijskih optimizacijskih problemov. Generično obliko algoritma lahko prilagodimo na način, da upošteva več optimizacijskih ciljev. Sposobnost preiskovanja različnih regij problemskega prostora omogoča iskanje raznorodnih rešitev.

Obstajata dva splošna pristopa k večkriterijski optimizaciji. Ena možnost je, da se posamezne kriterijske funkcije z uteženo vsoto združijo v eno samo funkcijo ali pa se vse kriterijske funkcije razen ene zapiše v obliki omejitev. V prvem primeru je vsaki komponenti treba določiti utež, kar utegne biti zahtevno, saj mora te uteži določiti ekspert. V praksi je tudi ekspertu na določenem področju težko natančno določiti uteži. Pri uporabi omejitev se srečamo s podobnim problemom, saj je omejitve treba ročno postaviti. V obeh primerih bo optimizacija vrnila eno najboljšo rešitev in ne več rešitev, ki upoštevajo kompromise med vsemi cilji [14]. Drugačen pristop je poiškati množico Pareto optimalnih rešitev ali njen približek. Pareto optimalna množica je množica rešitev, ki so med sabo enakovredne oziroma nedominirane. Pri primerjavi rešitev iz te množice opazimo, da vsaka rešitev za doseglo boljše vrednosti enega kriterija žrtvuje vrednost ostalih kriterijev. Pareto optimalne rešitve so pogosto zaželeno, saj ne dobimo zgolj ene same rešitve, ampak več kompromisnih rešitev. Število rešitev ni določeno, ampak se običajno povečuje s številom ciljev [14].

V diplomskem delu smo večkriterijsko optimizacijsko nalogo upravljanja pametne stavbe reševali z evolucijskim algoritmom. Razvoj algoritma je potekal na Odseku za inteligentne sisteme Instituta "Jožef Stefan" [15] v okviru projekta s podjetjem Robotina, ki je vodilni ponudnik na področju tehnologij nadzora sončnih elektrarn, trajnostnih stavb, sistemov obnovljivih virov energije in industrijske avtomatizacije. Cilj projekta je optimizirati upravljanje energetske učinkovitih pametnih stavb. Predlagana rešitev naj

bi omogočala energetska in stroškovno učinkovitost stavb, hkrati pa mora upoštevati uporabnikovo udobje. Za optimizacijo delovanja stavbe je predlagan večkriterijski evolucijski algoritem, ki upošteva kriterija energetske učinkovitosti in uporabnikovega udobja [16]. Projekt je služil kot študija primernosti uporabe evolucijskih algoritmov za optimizacijo delovanja pametnih stavb. Rešitev vsebuje spremljanje uporabnikov in učenje njihovih navad (prisotnosti), kar lahko uporabi optimizacijski algoritem. Rezultati optimizacije so urniki delovanja stavbe, ki jih ponudimo uporabniku preko uporabniškega vmesnika, ki omogoča preprosto interakcijo s sistemom.

Diplomsko delo je razdeljeno na pet poglavij. Uvodu sledi poglavje Problem optimizacije pametnih stavb, v katerem sta predstavljena teoretična osnova in podroben opis obravnavanega problema. V tretjem poglavju je opisana uporabljena metodologija. Poglavje se deli na tri podpoglavja. Prvo je podpoglavje Evolucijski algoritem, kjer je podrobno opisano delovanje algoritma, sestava kromosoma in genetskih operatorjev. Sledi podpoglavje Stavba, kjer sta opisana pametna stavba in način njenega modeliranja, v ločenih podpoglavjih pa tudi predstavljena izvedba ljudem prijaznih sprememb in strategije upravljanja z električno energijo. Temu sledi opis arhitekture sistema, postopka zagona optimizacije in pridobivanja podatkov za analizo. V četrtem poglavju so predstavljeni dobljeni rezultati, njihova podrobna analiza in kritično vrednotenje. V petem poglavju povzamemo opravljeno delo. Predstavljeni so glavni zaključki in ideje za izboljšave ter možnosti za izvedbo podobnega sistema v realnem okolju.

Poglavje 2

Problem optimizacije pametnih stavb

2.1 Teoretična osnova

Pametne stavbe so se začele razvijati v devetdesetih letih prejšnjega stoletja. Primarni cilj razvoja je bil avtomatizacija domov in gospodinjskih del. Pametna stavba vključuje napredno avtomatizacijo sistemov, ki prebivalcem zagotavljajo spremljanje in nadziranje delovanje stavbe. Funkcionalnosti pametnih stavb izvaja več sistemov, ki so med sabo povezani in so sposobni medsebojne komunikacije. Pametna stavba lahko nadzoruje notranjo razsvetljavo, predvajane multimedijske vsebine, varnost, obnašanje vhodnih vrat, temperaturo, kroženje in tok zraka v prostorih in mnoge druge funkcionalnosti [17]. Cilj pametnih stavb je uporabnikom zagotoviti čim večje udobje, energetska učinkovitost in varnost. Za delovanje večih sistemov hkrati je potrebno veliko število raznovrstnih naprav, vsaka s specifično vlogo. Elemente sistema praviloma delimo na pet tipov:

- senzorji, ki zajemajo podatke v stavbi in jih sporočajo sistemu (npr. senzor kakovosti zraka, temperature, prisotnosti in osvetljenosti);
- kontrolerji so tipično programi, ki kontrolirajo oziroma nadzirajo delo-

vanje stavbe. Zaznavajo odstopanja vrednosti, ki jih posredujejo senzorji in izdajajo navodila akterjem;

- akterji so naprave v stavbi, ki dejansko izvedejo želene spremembe (npr. toplotna črpalka);
- vodila;
- uporabniški vmesniki, preko katerih uporabnik lahko vidi trenutno stanje stavbe, spreminja nastavitve ipd.

Običajna pametna stavba ni posebej pametena, saj praviloma le sledi uporabnikovim pravilom in željam. Resnično pametna stavba bi se morala znati učiti iz zgodovine uporabnikovega preteklega obnašanja in se prilagajati spremembam le tega, hkrati pa iskati boljše načine za upravljanje doma. Predvsem pri vodenju stanovanjskih stavb se je pokazala zahteva po ustreznem obravnavanju uporabnikovih potreb sočasnega doseganja energetske varčnosti in visoke stopnje udobja. Zmanjševanje stroškov shranjevanja in obdelave podatkov, dostopnost senzorjev in aktuatorjev ter enostavno povezovanje različnih naprav v skupen sistem, omogoča uporabo kompleksnih metod vodenja tudi v manjših bivalnih enotah. Obstoječi sistemi pametnih hiš omogočajo avtomatizacijo delovanja sistemov v stavbah po vnaprej nastavljenih urnikih, preklapljanje med načini delovanja glede na zaznano prisotnost uporabnika ali na zahtevo uporabnika preko uporabniškega vmesnika. Večini uporabnikov ne uspe nastaviti primerne (optimalne) urnika za avtomatizacijo, saj morajo pri tem nastaviti veliko pogosto nerazumljivih parametrov in upoštevati nenehne spremembe svojih potreb in zunanjih vplivov. Poleg tega take rešitve praviloma ne izrabijo celotnega potenciala, ki jih sistemi hišne avtomatizacije omogočajo.

Za reševanje problema zagotavljanja ustreznega upravljanja pametne stavbe, ki zadovoljuje nasprotujoče si kriterije, je možna uporaba tehnik strojnega učenja za prepoznavanje navad uporabnikov in gradnjo napovednih modelov njihovega obnašanja ter uporabo večkriterijske optimizacije [18]. Problem

optimalnega vodenje sistemov HVAC, je zaradi številnih dejavnikov zelo zahteven. Glavne probleme pri tem predstavlja večdimenzionalna in nelinearna narava sistemov, ki je odvisna od uporabnikovih nastavitev in trenutnega časa v dnevu in z njimi povezane motnje v sistemu. Težave je tudi slabša kakovost in zanesljivost podatkov, kar je posledica pretvorbe analognih signalov senzorjev, ki se lahko obnašajo zelo nepredvidljivo, v digitalne signale, ki jih lahko uporabljamo v računalniških sistemih. Probleme predstavlja tudi uporabnikova interakcija, saj lahko uporabnik v katerem koli trenutku prekine trenutno avtonomno delovanje stavbe in preko uporabniškega vmesnika vnese svoje nove želje, ki morajo imeti prednost. Zaradi preprostosti so zelo razširjeni starejši in preprosti sistemi za upravljanje, kot sta dvopoložajna regulacija in PID-regulacija [19]. Naprave, ki uporabljajo dvopoložajno regulacijo, začnejo delovati, ko zaznajo preveliko odstopanje regulirane vrednosti in se izklopijo, ko je cilj dosežen. PID-regulacija upošteva trenutno odstopanje stanja sistema od želenega ter tudi seštevke prejšnjih odstopanj in napovedano odstopanje glede na trenutno dinamiko sistema.

Cilj optimalnega vodenja je poiskati strategijo, ki reši optimizacijski problem pri neki kriterijski funkciji. Kriterijska funkcija lahko upošteva minimizacijo porabe energije, minimizacijo prizadevanja za vodenje, maksimizacijo udobja ali pa celo več kriterijskih funkcij hkrati. Poleg tega upošteva še podane omejitve sistema in zakonska določila pri upravljanju z električno energijo (predvsem regulacija prodaje električne energije). Različne zahteve uporabnikov pametnih stavb si med seboj pogosto nasprotujejo: potrebno je zagotavljati udobje, obratovalni stroški morajo biti čim nižji, v zadnjem času pa se čedalje pogosteje izraža tudi zahteva po energetski učinkovitosti in okoljski sprejemljivosti. Primer problema več nasprotujočih si dejavnikov lahko ponazorimo s preprostim primerom reguliranja temperature vode in delovanje črpalke. Maksimalno udobje bi zahtevalo neprestano razpoložljivost tople vode, kar ni stroškovno najučinkoviteje. Stalna razpoložljivost tople vode zahteva neprestano dogrevanje in delovanje toplotne črpalke, kar je energetsko potratno. Kriterij energetske učinkovitosti stremi k temu, da se voda

seGREJE tik pred uporabnikovo uporabo. V kolikor bi znali predvideti oziroma napovedati obnašanje uporabnikov s sprotnim učenjem njihovih življenjskih navad, bi za enako stopnjo udobja lahko porabili bistveno manj energije. Sistemi v pametnih hišah ne morejo samostojno odločati o optimalnih vrednost temperature prostorov, osvetljenosti ali temperature vode, lahko le spremljajo uporabnike. Optimizacijski algoritmi praviloma ne predvidevajo možnosti, da bi se uporabnik odpovedal določenemu udobju v zameno za nižjo porabo energije.

2.2 Obravnavani problem

Zahteven problem pri upravljanju notranjega bivalnega okolja je, da sta uporabnikovo udobje in energetska poraba v medsebojnem konfliktu. Za zagotavljanje maksimalnega udobja je potrebno veliko energije in s tem povezanih veliko stroškov, ki jih ne želimo. Več možnosti pri odločanju lahko ponudimo uporabniku, če se za problem s tema dvema ciljema najde Pareto optimalna fronta, ki jo sestavljajo kompromisne rešitve med udobjem in stroški. Problem je matematično formuliran v nadaljevanju [20]. Predpostavimo, da je trenutno stanje temperatur v hiši $T_{\text{zacetna}} = T_0$, ukazi, ki se uporabljajo za upravljanje z akumulatorjem, $A_{\text{zacetni}} = A_0$, in časi, ob katerih se zaženejo porabniki v hiši, $P_{\text{zacetni}} = P_0$. Za doseg optimalnih rešitev so izbrane spremenljivke zelena temperatura T , zeleni ukazi upravljanja z akumulatorjem A in zeleni začetni časi porabnikov P . En cilj je seveda minimizirati neudobje, drugi pa zmanjšati skupne stroške:

$$\min(\text{neudobje}) = \min(1 - \text{udobje}) \quad (2.1)$$

$$\min(\text{Energija}_{\text{porabljen}}) = \min(E_1(T) + E_2(A) + E_3(P)) \quad (2.2)$$

kjer so $E_1(T)$, $E_2(A)$, $E_3(P)$ energija porabljen za zagotavljanje vrednosti T , A in E . Vrednosti $E_1(T)$, $E_2(A)$, $E_3(P)$ so pridobljene z uporabo simulatorja modela hiše.

Cilj algoritma je, da najde boljše ali pa vsaj enako dobre rešitve, kot je začetni urnik, ki ga je vnesel uporabnik kot svojega idealnega. Pričakujemo

lahko več rešitev, ki imajo manj udobja zaradi odstopanja od želene temperature in s tem povezane nižje stroške. Izključena je še dražja in udobnejša rešitev od začetnega urnika, saj ta predstavlja maksimalno uporabnikovo udobje, zato se bo začetni urnik vedno nahajal med rešitvami, razen če algoritem najde rešitev, ki dosega isto udobje z manj stroški, torej ga dominira. Stavba je opremljena s fotovoltaičnima paneloma, ki ob sončnem obsevanju proizvajata električno energijo. Za napajanje vseh električnih naprav v stavbi sta fotovoltaična panela primarni cilj energije. Stavba je opremljena z akumulatorjem, kamor je možno proizvedeno energijo shranjevati. Pri ceni določenega urnika ima veliko vlogo upravljanje z akumulatorjem. S pametnim upravljanjem energije, v kar vključujemo prodajo in shranjevanje odvečne energije ob sončnem vremenu, lahko veliko privarčujemo z minimalnim ali celo brez negativnega učinka na uporabnikovo udobje. Na shranjevanje odvečne energije je vezano tudi obnašanje hiše v oblačnih dneh, ko ni presežka sončne energije. Takrat je pomembno, ali moramo energijo črpati iz omrežja ali pa napajamo hišo iz akumulatorja z energijo, ki smo jo prej vanj shranili. Razporeditev delovanja porabnika v optimizacijskem oknu je pomembna odločitev. Razlike se namreč pojavijo pri zagotavljanju energije za delovanje porabnika. Možnih je več scenarijev:

- energijo za delovanje porabnika je potrebno črpati v času višje tarife,
- energijo za delovanje porabnika je potrebno črpati v času nižje tarife,
- porabnika lahko neposredno napajamo z energijo neposredno iz fotovoltaičnih panelov,
- porabnika napajamo iz energije shranjene v akumulatorju, pri čemer moramo upoštevati energetska izgubo pri shranjevanju in črpanju. Ta znaša 10% in je ključnega pomena pri vprašanju kam z energijo, ki jo fotovoltaični paneli proizvedejo, stavba pa je v trenutku proizvodnje ne porabi;

Pojavi se vprašanje, ali električno energijo prodati ali jo shraniti v akumulator. Odgovor ni enostaven, saj je potrebno upoštevati veliko dejavnikov,

kot so vremenska napoved, cena prodaje energije in napoved uporabnikovih aktivnosti ter s tem povezano porabo energije.

Uporabnikovo udobje je izračunano kot odmik dejanske temperature od želene temperature, ki jo je vnesel uporabnik. Udobje se meri samo v obdobjih, ko je uporabnik prisoten v stavbi.

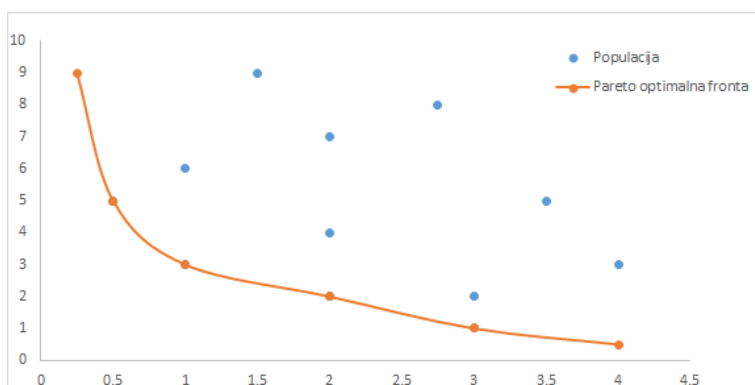
Energetska poraba stavbe za segrevanje, hlajenje in poganjanje porabnikov se na podlagi cene energije pretvori v stroške z upoštevanjem cene odvzema energije v času višje in nižje tarife ter vrednost energije, ki ostane na koncu optimizacijskega obdobja shranjena v akumulatorju. V primeru, da ne upoštevamo shranjene energije, algoritem ne bi shranjeval energije v akumulator, saj bi se ta enačila z zavrženo energijo. Energija je ovrednotena enakovredno kot energija v času višje tarife. Stroške izračunamo tako, da beležimo vsako izmenjavo energije z omrežjem in količino energije pomnožimo z ustrezno mero. V primeru odvzema energije je ta vrednost bodisi vrednost višje bodisi nižje tarife, odvisno od časa odvzema. Ob prodajanju energije v omrežje se količina oddane energije pomnoži s prodajno ceno. Na koncu se dobiček od prodaje odšteje od stroškov, ki so rezultat odvzema energije. Vse omenjene cene energije, razen vrednosti energije v akumulatorju, ki jo lahko ovrednotimo sami, določa elektrodistributer in nanje nimamo vpliva.

Poglavje 3

Uporabljene optimizacijske metode

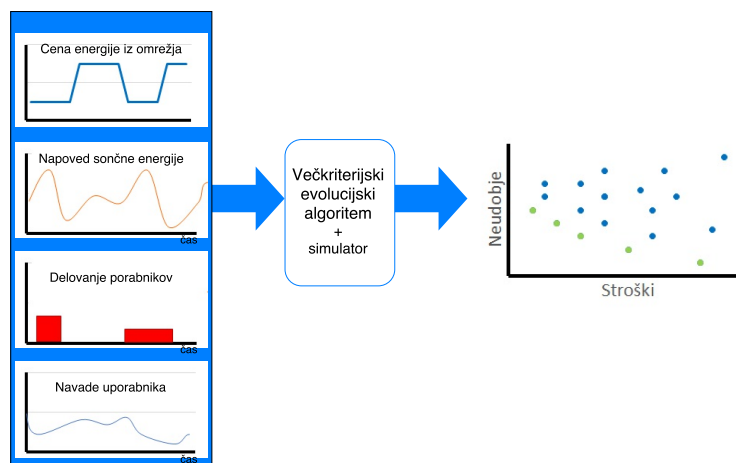
3.1 Evolucijski algoritem

Uporabili smo evolucijski algoritem, prilagojen za večkriterijsko optimizacijo. Kriterija optimizacije sta bila maksimizacija uporabnikovega udobja in minimizacija stroškov vodenja. Problem maksimizacije uporabnikovega udobja smo prevedli v minimizacijo neudobja. Pri večkriterijskih problemih so cilji, ki jih želimo optimizirati ponavadi v medsebojnem konfliktu, zato ni možna samo ena optimalna rešitev. Za tak tip problemov je namesto ene same rešitve smiselno najti množico kompromisnih rešitev ali t.i. Pareto optimalnih rešitev, ki v prostoru kriterijev tvorijo Pareto optimalno fronto. Slika 3.1 prikazuje primer Pareto optimalne fronte. Vsaka rešitev na njej je nedominirana, kar pomeni, da zanjo ne obstaja rešitev, ki bi imela oba kriterija boljša od nje. Vsaka rešitev ima od vseh ostalih natanko eno komponentno boljšo, drugo pa slabšo.



Slika 3.1: Pareto optimalna fronta.

Optimizacijski algoritem kot vhodne podatke uporabi podani začetni (ne-optimirani) urnik stavbe in podatke o ceni energije, napovedi sončne energije, porabnikov v sistemu in navadah uporabnikov. Vrednotenje novo generiranih rešitev poteka s simulatorjem, ki za podani urnik vrne uporabnikovo udobje in stroške upravljanja. Potek delovanja algoritma je prikazan na sliki 3.2.



Slika 3.2: Shema delovanja algoritma.

Dokler ni izpolnjen ustavitveni pogoj, poteka preiskovanje prostora urnikov in njihovo vrednotenje preko omenjene simulacije. Uporabniku se preko grafičnega vmesnika ponudi Pareto optimalne rešitve, med katerimi lahko izbere poljubno. Optimizacijsko okno, znotraj katerega izvajamo optimizacijo je dolgo 48 ur. Pseudokoda algoritma je sledeča:

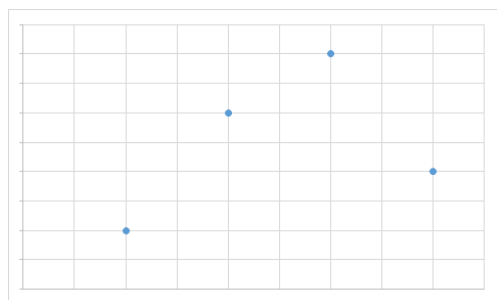
```
Pripravi množico doslej znanih rešitev S
Generiraj in ovrednoti začetno populacijo P
Dodaj populacijo P v množico S
Dokler ni izpolnjen ustavitveni pogoj, ponavljaj
  Pripravi prazno populacijo potomcev D
  Ponovi p/2krat
    S turnirsko selekcijo izberi dva starša iz P
    Starša križaj, da dobiš dva potomca
    Potomca mutiraj in ovrednoti
    Potomca dodaj v S
  Najboljša izmed staršev in potomcev dodaj v D
Populacija potomcev D postane populacija staršev P
Vrni nedominirane rešitve iz množice S
```

Podrobnosti in integracija algoritma z izbranim agentnim okoljem bo opisana kasneje.

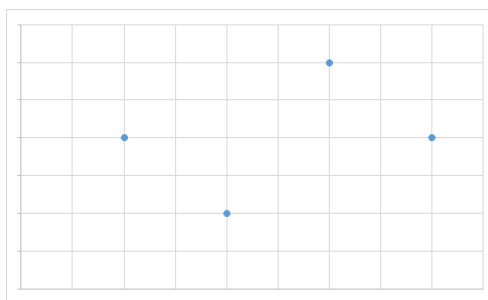
Začetni urnik je določen na podlagi vremenske napovedi, uporabnikovih navad in njegovih temperaturnih želja. Nastavitve temperature začetnega urnika se popolnoma ujemajo s temperaturami, ki jih uporabnik nastavi kot idealne temperature notranjega okolja hiše. Energetski komponenti začetnega urnika se izračunata glede na predvidevanja o energetskem stanju stavbe. Predstavitev urnika (kromosoma) bo opisana v podpoglavju 3.1.1. Prilaganje temperature želeni temperaturi bo pomenilo veliko udobja (malo neudobja), seveda to privede tudi do višjih stroškov, saj je potrebno za vsak interval vzdrževati točno določeno idealno temperaturo.

Zaradi postopka dodajanja osebkov v populacijo potomcev D (glej pseudokodo algoritma) se lahko zgodi, da populacija D ni enako velika (ne vsebuje

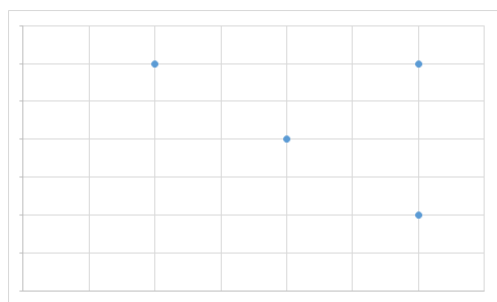
enakega števila osebkov) kot populacija P. Razlog se skriva v določanju najboljših izmed staršev in obeh potomcev (psevdokoda, vrstica 11). Izmed množice štirih osebkov moramo izbrati dva, kar ni vedno mogoče. Pri vrednotenju štirih osebkov so možni scenariji prikazani na sliki 3.3, na kateri oba kriterija minimiziramo.



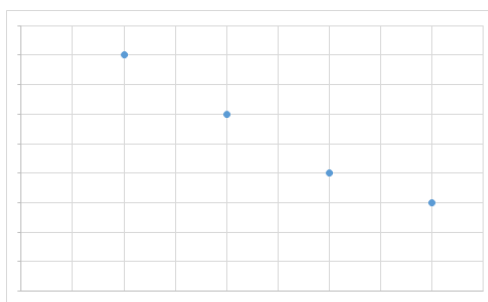
(a) En osebek dominira tri.



(b) Dva osebka dominirata dva.



(c) Trije osebki dominirajo enega.

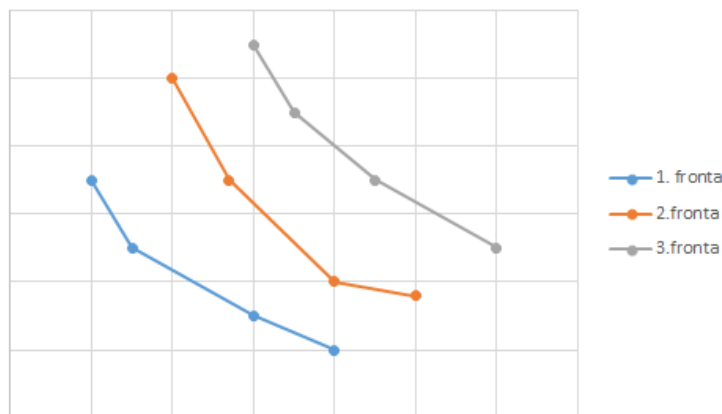


(d) Vsi osebki nedominirani.

Slika 3.3: Možni scenariji pri dodajanju osebkov v populacijo potomcev.

V primeru prikazanem na sliki 3.3(b) nimamo težav z izbiro dveh najboljših primerkov, saj sta edina nedominirana. V primerih na sliki 3.3(c) in sliki 3.3(d)) imamo več kompromisnih rešitev, med katerimi ne moremo določiti dveh najboljših. V populacijo D se tako dodajo vsi osebki na prvi (najboljši) fronti. V takem primeru bo populacija potomcev D večja kot populacija P. Rešitev bo opisana kasneje. V primeru prikazanem na sliki 3.3(a) je zgolj en osebek nedominiran, zato se bo dodal v populacijo D. Problem, kako izmed ostalih treh izbrati še enega, lahko rešimo s principom front.

V splošnem ja fronta skupina osebkov, ki jo dominira isto število osebkov. Splošen primer front prikazuje slika 3.4.



Slika 3.4: Prikaz front.

V primeru na sliki 3.3(a) nedominirani osebek predstavlja prvo fronto. Ostali trije se nahajajo na isti fronti in za izbiro se uporabijo medsebojne razdalje. Zaradi raznovrstnosti rešitev in nadaljnjih možnosti pri preiskovanju prostora le-teh, si želimo v populaciji karseda različne rešitve. Želeno obnašanje dosežemo, če za vsako točko izračunamo razdaljo do njenega najbližjega sosedu. Izberemo tistega, ki je najbolj oddaljen od ostalih dveh. Če se na drugi fronti nahajata dva izmed preostalih treh osebkov, imata medsebojno razdaljo enako. Takrat izberemo naključnega. Opisan pristop deluje po principu delnega elitizma, kjer je potomčeva edina možnost preživetja ta, da ga starš ne dominira. Možnosti sta dve in sicer bodisi potomec dominira starša, bodisi potomec predstavlja kompromisno rešitev in je za izboljšanje ene komponente žrtvoval drugo.

Kljub dejstvu, da je možno vedno izbrati natanko dva nedominirana osebkov iz skupine obeh staršev in potomcev, nam to ne zagotavlja, da se bodo vse rešitve v populaciji D nahajale na isti fronti. Nekateri osebki (v nekaterih skrajnih primerih tudi en sam osebek) lahko dominira celotno preostalo populacijo potomcev D. V primeru, da je na koncu iteracije popula-

cija potomcev D večja od velikosti populacije P, je potrebno populacijo D zmanjšati. Ta problem rešimo z zgoraj opisanim pristopom. Osebke razporedimo v fronte in polnimo populacijo z najboljšimi osebki. Izbiranje lahko ponazorimo s psevdokodo:

```

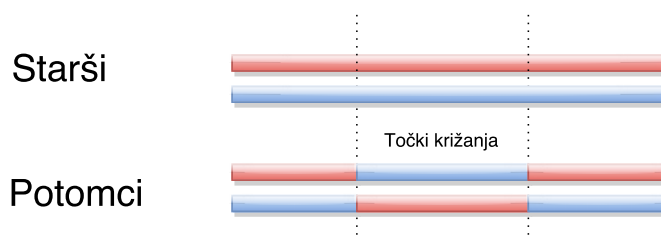
št_mankajočih = 0
populacija P = prazna množica
// p – velikost populacije: parameter algoritma
št_mankajočih < p:
    fronta = najdi 1. fronto v populaciji D
    if( velikost(fronta) < št_mankajočih ):
        dodaj vse iz fronte v P
        odstrani vse iz fronte iz D
        št_mankajočih += velikost(fronta)
    else:
        izberi št_mankajočih iz fronte
        izbrane dodaj v P

```

Evolucijski algoritmi zahtevajo parametre, ki določajo, kakšne so verjetnosti posameznih dogodkov oziroma pojavov med evolucijo populacije. Uporabljeni parametri in njihove vrednosti so naslednji:

- verjetnost mutacije: 3%; pri generiranju potomčevega dednega materiala so možne majhne mutacije za vsak posamezen gen;
- velikost populacije: 10 osebkov;
- velikost turnirja: 3;
- verjetnost križanja: 70%; za vsako komponento kromosoma staršev se s to verjetnostjo istoležna komponenta potomčevega kromosoma določi s križanjem;
- število točk križanja: 2. Križanje je pomembna operacija, ki omogoča, da vsak od staršev da potomcu določen del dednega zapisa. Primer

kako starša z uporabljenim dvomestnim križanjem preneseta svoj zapis na potomca je prikazan na sliki 3.5.



Slika 3.5: Prikaz delovanja križanja.

Rdeča in modra barva predstavljata dedni zapis vsakega od staršev. S točkama križanja, ki se izberejo naključno, je določeno kakšen dedni zapis bosta podedovala potomca.

- Ustavitveni pogoj: 1000 pregledanih rešitev. Pogosto si pri evolucijskih algoritmi želimo maksimalno izkoristiti čas, ki ga imamo na voljo. Zaradi časovnih omejitev in dolgotrajnosti vrednotenja vsake rešitve smo uporabili 1000 pregledanih rešitev oziroma 100 generacij. Algoritem je za dokončanje svoje naloge v povprečju potreboval okoli 100 minut.

Implementacija evolucijskega algoritma ni možna brez generatorja naključnih števil, ki določa izid naključnih dogodkov. Vsakemu generatorju naključnih števil je potrebno določiti seme, saj resnično naključnih rezultatov v računalniškem svetu ni. Seme dejansko določa izid generiranja naključnih števil, saj vsak generator psevdo naključnih števil potrebuje začetno število, ki služi kot osnova za izračun vseh ostalih. Za generiranje generatorja psevdo naključnih števil je bil uporabljen `java.util.Random` [21].

Začetna populacija je ustvarjena z usmerjenim naključnim pristopom. Velikost populacije deset pomeni, da je poleg začetnega potrebno generirati še devet urnikov. Pet osebkov prvotne generacije je vedno generiranih s preturbiranjem začetnega urnika. Preostali štirje urniki so rešitve prej izvedenih optimizacij. Želeni učinek je pospešitev delovanja algoritma, hkrati

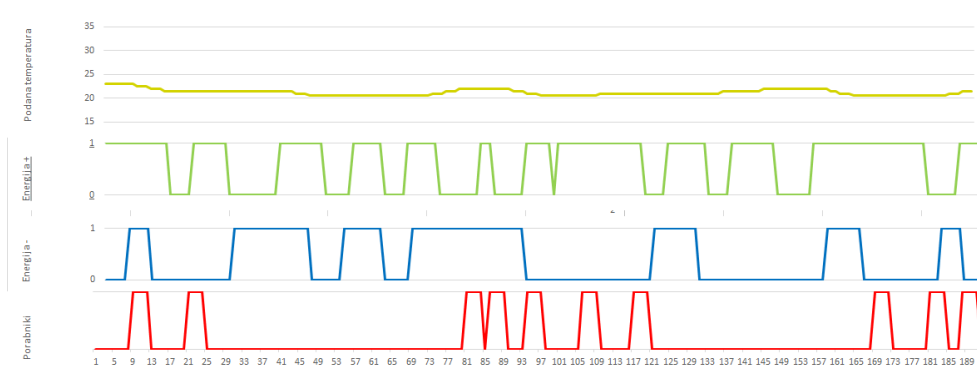
pa omogočanje novih, različnih rešitev. Optimizacijski agent si zapomni vse začetne urnike in dosežene rešitve pri določenem začetnem urniku. Različni vremenski pogoji bodo vodili v različne začetne urnike in rešitve. Za smiselnost tako dobljenih urnikov med vsemi začetnimi urniki poiščemo tistega, ki je trenutnemu začetnemu urniku najbolj podoben.

3.1.1 Kromosom

Vsak urnik opisuje delovanje stavbe med podanima začetnim in končnim časom, pri čemer je vmesno obdobje razdeljeno na 15-minutne intervale. Urnik je sestavljen iz štirih komponent:

- temperatura: za vsak interval določimo želeno temperaturo v °C, ki se mora nahajati znotraj intervala $[T_{\min}, T_{\max}]$, kjer lahko T_{\min} in T_{\max} določimo za vsak posamezen interval posebej;
- energija+: v primeru, da fotovoltaični paneli proizvedejo več energije, kot jo stavba porabi za svoje delovanje, za vsak interval določimo stanje akumulatorja; možni sta vrednosti 1, kar pomeni, da naj se akumulator polni in 0, kar pomeni, da naj se akumulator ne polni;
- energija-: v primeru, da fotovoltaični paneli proizvedejo manj energije, kot jo stavba porabi za svoje delovanje, za vsak interval določimo stanje akumulatorja; možni sta vrednosti 1, kar pomeni, da naj se akumulator prazni in 0, kar pomeni, da naj se akumulator ne prazni.
- porabniki: za vsakega porabnika v stavbi določimo začetni interval delovanja; končni čas in količina porabljene energije sta izračunana na podlagi lastnosti porabnika; za vsak interval je tako možna vrednost 1, kar pomeni, da je porabnik aktiven in 0, kar pomeni, da porabnik ni aktiven.

Primer urnika, ki je na ta način zapisan v kromosomu, je prikazan na sliki 3.6.

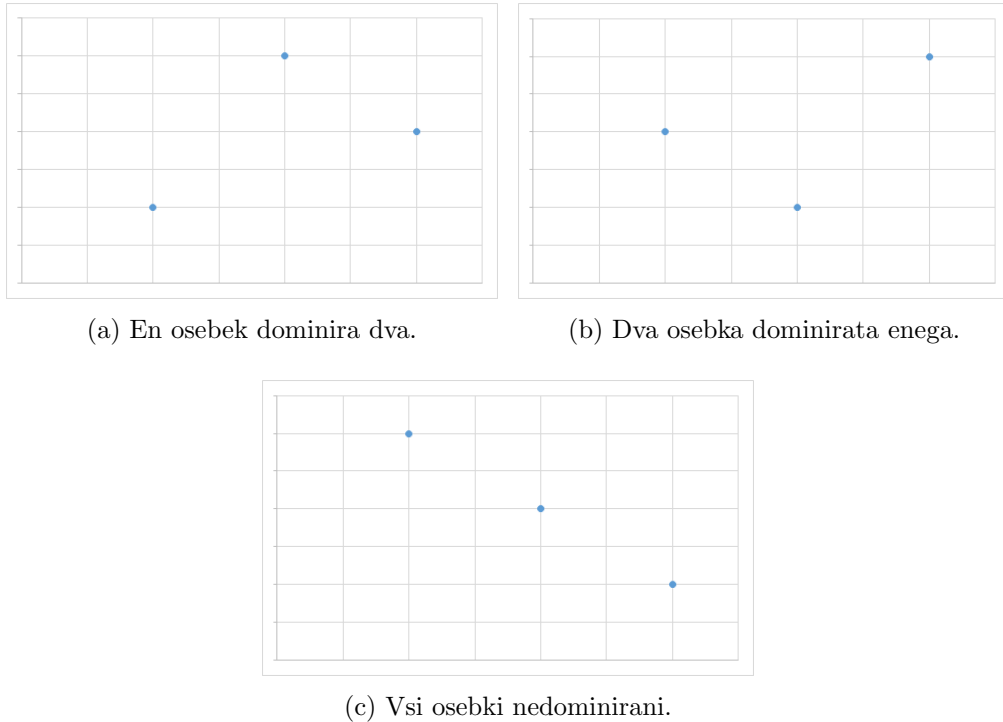


Slika 3.6: Primer urnika.

3.1.2 Genetski operatorji

Predlagana definicija kromosoma pomeni, da je klasične genetske operatorje selekcije, križanja in mutacije potrebno spremeniti oziroma jih prilagoditi. Zaradi različne strukture komponent kromosoma se genetske operacije pri različnih komponentah kromosoma razlikujejo.

- Turnirska selekcija: iz populacije so bili izbrani trije naključni osebki. Postopek izbiranja ponovimo dvakrat, z upoštevanjem dejstva, da zmagovalca prvega turnirja ne moremo ponovno izbrati, torej pri velikosti populacije P vzorčimo med $P - 1$ osebki. Določanje najboljšega med tremi osebki po več kriterijih lahko predstavlja problem, saj ni vedno en osebek dominanten. Možne situacije so predstavljene na sliki 3.7, kjer oba kriterija minimiziramo.



Slika 3.7: Možni scenariji v turnirski selekciji.

Problem rešimo na podoben način, kot v prej opisanem problemu neujemanja velikosti populacije D . V primeru na sliki 3.7(a)) ni težav, saj en osebek dominira ostala dva. V primeru na sliki 3.7(b), kjer sta dva osebka nedominirana, izberemo naključnega. V primeru nedominiranosti vseh osebkov na sliki 3.7(c), kjer se nahajajo vsi na isti fronti, izberemo tistega z največjo razdaljo do ostalih dveh.

- Križanje: uporabljena je bila klasična metoda križanja, kot je opisano v poglavju 3. Križanje temperaturnih vrednosti in obeh energetskih komponent, ki narekujejo upravljanje z akumulatorjem, poteka z n -točkovnim križanjem, kjer je n parameter algoritma. Pri križanju porabnikov je uporabljeno enotočkovno križanje, kjer se pri istoležnem porabniku križata začetna intervala delovanja.
- Mutacija: najlažja je uporaba mutacije pri komponentah Energije+ in

Energije–, saj sta možni vrednosti le 1 in 0. Ob mutaciji se vrednosti preprosto zamenjata, 1 torej postane 0 in obratno. Mutacija temperature predstavljajo večjo težavo in zahteva nekatere omejitve. Ni smiselno dopustiti spremembe temperature na poljubno vrednost iz zaloge vrednosti realnih števil. Zalogo vrednosti, iz katere lahko mutacijski operator črpa, lahko določimo za vsak interval posebej. Nastavljene so bile smiselne vrednosti za zgornjo in spodnjo mejo. Zgornja meja je bila postavljena na 24, spodnja pa na 17 °C. Mutacija porabnikov poteka na razmeroma enostaven način. V 48-urnem optimizacijskem oknu je 196 15-minutnih intervalov, a vsi niso primerni za začetek delovanja porabnika, ker se mora celotno delovanje zaključiti znotraj optimizacijskega okna. Določimo začetni interval, kjer so možni intervali definirani: $[0, 196 - \text{DelovanjePorabnikaV15minutnihIntervalih})$.

3.2 Stavba

Za modeliranje stavbe in simulacije je bil uporabljen program EnergyPlus [22]. EnergyPlus je orodje, ki je sposobno, na podlagi uporabnikovega fizičnega opisa stavbe in naprav v njej, izračunati obremenitve ogrevanja in hlajenja, potrebne za ohranitev želene temperature. Stavba je bila modelirana po vzoru realnega kontejnerja in zato obsega zgolj 1 prostor, ki vsebuje:

- fotovoltaična panela (6×240 W), ki sta zmožna iz sončnega obsevanja proizvajati električno energijo;
- akumulator s kapaciteto 5 kWh. Izguba ob črpanju energije iz akumulatorja znaša 10%. Polnjenje je možno z močjo do 2,5 kW, črpanje pa do 5 kW;
- toplotna črpalka, ki skrbi za segrevanje oziroma ohlajanje stavbe;
- porabnik, ki predstavlja energetskega porabnika z razmeroma kratkim, a energetsko zahtevnim delovanjem, ki se nahaja v vsaki stavbi (na primer pomivalni ali pralni stroj);

- senzor temperature;
- senzor prisotnosti;
- vremenska postaja;
- električno omrežje: Gre za standardno električno omrežje, v katerega je stavba priključena in omogoča odvzem elektrike, kadar fotovoltaična panela ne uspeta ustvariti zadostne količine, ter prodajanje elektrike, kadar si tega želimo.

Konfiguracija stavbe je shranjena v datotekah xml in idf, iz katerih EnergyPlus ob zagonu prebere parametre stavbe in iz njih zgradi model. Definiran je sledeč prostor:

- širina: 1,85 m,
- dolžina: 6,73 m,
- višina: 3,62 m,
- dva okni velikosti $1,3 \times 0,81$ m,
- vrata dimenzij: $1,07 \times 2,5$ m.

Debelina sten, oken in vrat ni resnično pomembna, saj je izolacijske lastnosti gradbenih elementov moč nastaviti v konfiguracijski datoteki xml.

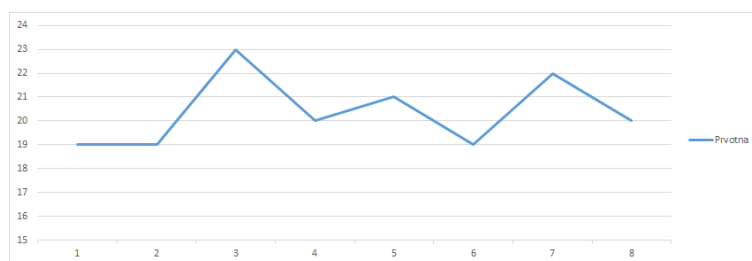
Na sliki 3.8 je prikazan prototip realnega okolja. Gre za kontejner, ki je postavljen poleg podjetja Robotina. V njem so nameščene zgoraj navedene naprave. Manjše prototipno okolje je idealno za potrebe eksperimentiranja, saj omogoča rešiti probleme povezljivosti in integracije ter prikazati morebitne prednosti, ki jo vključitev optimizacijskega algoritma lahko prinese v sistem hišne avtomatizacije.



Slika 3.8: Prototip pametne stavbe.

3.2.1 Izvedba ljudem prijaznih sprememb

Pri ustvarjanju novih urnikov s križanjem in mutacijo lahko pride do velikih razlik, na primer temperaturnih vrednosti med dvema intervaloma. Primer tako generirane temperature prikazuje slika 3.9.



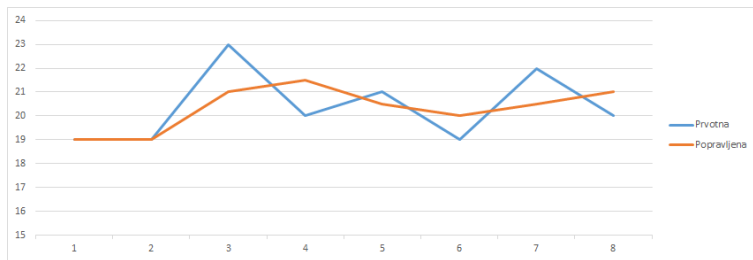
Slika 3.9: Primer poteka temperature, dobljenega s križanjem.

Če bi stavba želela točno slediti takim navodilom, bi bilo v intervalu T_3 potrebno močno segrevanje prostora, v intervalu T_4 pa spet močno ohlajanje, saj bi stavba morala doseči spremembo temperature za 4 oziroma 5 °C v časovnem obdobju 15 minut. Ljudem tako močno hlajenje oziroma gretje praviloma ni všeč in ne prinaša udobja.

Hkrati vemo, da si uporabnik v intervalih do T_2 želi $19\text{ }^{\circ}\text{C}$ (na primer med spanjem), v intervalu T_2 pa se zbudi in si želi temperaturo 23 . Izkaže se, da je problem izbire strategije segrevanja stavbe težak. Če želimo zagotoviti 19 stopinj v T_2 in 23 v T_3 , pomeni, da bomo sicer zadostili uporabnikovim temperaturnim željam, a jih bomo dosegli na neudoben način. Odločili smo se za strategijo glajenja temperaturnih vrednosti T in sicer, vsaka vrednost popravljene temperature v intervalu t se izračuna po naslednji formuli:

$$T_{\text{popravljena}}(t+1) = 0,5 * T(t+1) + 0,5 * T(t) \quad (3.1)$$

Vrednosti nato zaradi ločljivosti reguliranja temperature na $0,5\text{ }^{\circ}\text{C}$ še zaokrožimo. Primer delovanja zgornjega primera, kot ga prikazuje slika 3.9, z uporabo te metode transformiramo v delovanje prikazano na sliki 3.10.



Slika 3.10: Primerjava prvotne in popravljene temperature.

Graf popravljene temperature na sliki 3.10 pokaže, kako lahko na preprost način preprečimo pretirano hlajenje oziroma gretje v kratkem časovnem intervalu.

3.2.2 Upravljanje z elektriko

Zakonske omejitve, ki veljajo v RS glede upravljanja z elektriko in domačo elektrarno:

- Elektrike ni moč prodajati v omrežje iz akumulatorja. Torej elektrike ne smemo ob sončnih dnevih shranjevati v akumulator in jo kasneje prodajati.

- Električne, ki jo črpamo iz omrežja, ne smemo shranjevati v akumulator. To zakonsko določilo prepreči scenarij, ko bi lahko čez noč (v času nižje tarife) kupovali elektriko, jo shranjevali v akumulator in bi jo kasneje (v času višje tarife) uporabljali.

Slovensko električno omrežje je glede tarif in cen v njem statično, kar pomeni, da sta v nekem časovnem obdobju časovni okni obeh tarif (višje in nižje) znani in se ne spreminjata, kakor tudi ne njihova cena odvzema. Ob predpostavki statičnosti cen odvzema, prodaje in časa tarif, ki v Sloveniji velja, je možno za vsak scenarij izračunati stroškovno najučinkovitejšo strategijo upravljanja z energijo. Potrebno je poznati:

- ceno odvzema elektrike v višji tarifi (VT),
- ceno odvzema elektrike v nižji tarifi (NT),
- ceno prodaje elektrike, ki je praviloma neodvisna od tarife, torej je enaka v višji in nižji tarifi (PC),
- stanje razpoložljivosti električne energije v stavbi, torej ali imamo višek energije (fotovoltaična panela proizvedeta več energije, kot jo stavba potrebuje) ali manjko energije (fotovoltaična panela proizvedeta manj energije, kot jo stavba potrebuje).

Z znanjem o lastnostih statičnega omrežja, ki jih pridobimo, je možno sestaviti tabelo, ki prikazuje strategijo upravljanja z energijo v takih omrežjih. Strategija je razdeljena na dva dela: prvi za čas višje tarife (tabela 3.1) in drugi za čas nižje tarife (tabela 3.2).

Višek energije oziroma manjko energije predstavljata razliko med proizvedeno energijo fotovoltaičnih panelov in porabo stavbe.

$Stanje = ProizvedenaEnergija - Poraba$. Če je $Stanje < 0$ pomeni, da imamo manjko energije. V primeru, da je $Stanje > 0$ imamo višek energije. V malo verjetnem scenariju, kjer bi fotovoltaična panela proizvedla ravno toliko energije, kot bi jo stavba porabila, stanja ni potrebno določiti, saj upravljanje z energijo ni mogoče.

Vrednosti 1 in 0 predstavljata stanja akumulatorja. Vrednost 1 pomeni, da je akumulator aktiven, kar v situaciji viška energije pomeni, da se akumulator polni, v času manjka energije pa to pomeni, da se energija črpa iz akumulatorja. Vrednost 0 pomeni, da akumulator ni aktiven. Med viškom energije to pomeni, da se energija prodaja v omrežje, med časom manjka energije pa to pomeni, da se energija črpa iz omrežja. Opisano predstavlja želeno obnašanje, ki zaradi nepredvidljivih sprememb v okolju ali obnašanju uporabnika, ni vedno izvedljivo. V času manjka energije in ukaza 1 (energija naj se črpa iz akumulatorja) se lahko zgodi, da je akumulator prazen. V tem primeru se energijo črpa iz omrežja. Podobno velja ob višku energije z ukazom 1 in polnem akumulatorju, saj se bo odvečna energija avtomatsko prodajala v omrežje.

Tabela 3.1: Strategija upravljanja z energijo v času višje tarife.

	$PC > VT$	$VT \leq PC \leq NT$	$PC < NT$
Višek energije	0	1	1
Manjko energije	0	1	1

V času višje tarife bo v večini akumulator aktiven, razen v primeru, da je prodajna cena večja od višje tarife, kar praviloma ni možno. Ob višku energije je energijo smiselno shranjevati, saj je energija shranjena v akumulator več vredna. V času manjka energije je, če le mogoče, smiselno energijo črpati iz akumulatorja, saj je odvzem energije v času višje tarife najdražji.

Tabela 3.2: Strategija upravljanja z energijo v času nižje tarife.

	$PC > VT$	$VT \leq PC \leq NT$	$PC < NT$
Višek energije	0	1	1
Manjko energije	0	0	1

V obdobju nižje tarife bo aktivnost akumulatorja raznovrstnejša, saj obnašanje ni povsem odvisno samo od razmerja med obema tarifama in prodajno ceno. V času viška energije, za katerega lahko predvidevamo, da ga ne

bo veliko, je stroškovno učinkoviteje shranjevati energijo, saj lahko pridobi na vrednosti, če jo uporabimo v času višje tarife namesto, da bi energijo črpali iz omrežja. Isti logiki sledi dejstvo, da tudi če imamo shranjeno energijo v akumulatorju, se nam je ne splača črpati med časom nižje tarife, saj je takrat odvzem energije najcenejši in se shranjeno energijo bolj splača porabiti v času višje tarife.

Podatki za Slovenijo v času izvajanja so bili sledeči:

- Višja tarifa: vsak delovnik med 6:00 in 22:00, cena odvzema: 0,08018 €/kWh.
- Nižja tarifa: med vikendi in vsak delovnik med 22:00 in 6:00, cena odvzema: 0,04393 €/kWh.
- Prodajna tarifa: je statična in se ne spreminja. Je uravnoteženo povprečje višje in nižje tarife ter znaša 0,05978 €/kWh.

Strategija je splošna in jo je moč uporabiti v vsakem omrežju, za katero smo prepričani, da se med simulacijo lastnosti električnega omrežja, na katero je stavba priključena, ne bodo spreminjale. Glavni razmisleki pri razvijanju strategije so bili naslednji:

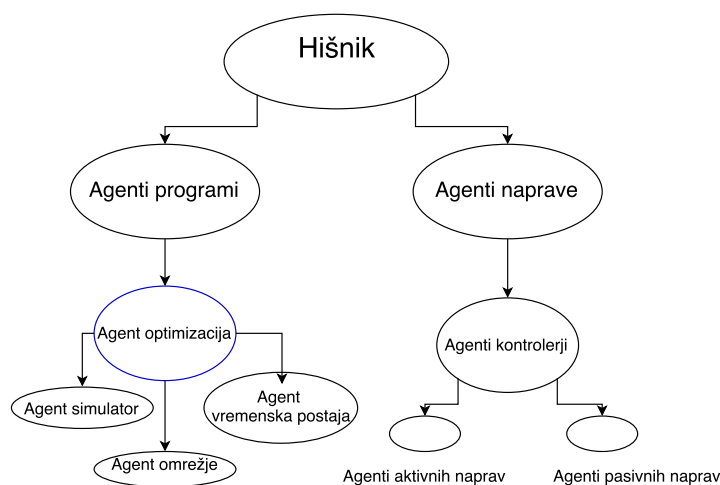
- V času nižje tarife (med 22:00 in 6:00) ne moremo pričakovati viška energije, saj ponoči ni sončnega obsevanja. Izjema se pojavi le v poletnih mesecih, ko sonce zgodaj vzide. Če želimo akumulator polniti, bo to potrebno storiti v času višje tarife.
- V primeru manjka energije, je nočemo kupovati v času višje tarife. Nima smisla kupovati drage energije, zato jo, če je možno, črpamo iz akumulatorja. V času nižje tarife energijo kupujemo, saj je energija v akumulatorju več vredna, ker jo bomo porabili v času višje tarife. Cena oziroma vrednost shranjene energije v akumulatorju je namreč večja od cene energije v nižji tarifi. Tako ne bo potrebno kupovati energije v času višje tarife.

3.3 Arhitektura sistema

Arhitekturo sistema je zasnovala razvojna ekipa Instituta “Jožef Stefan” v sodelovanju s podjetjem Robotina. Arhitektura sistema je definirana s hierarhično urejenim večagentnim sistemom [23]. Vsak agent ima določene zahtevane vhodne podatke, akcije, ki jih lahko izvaja, in izhodne podatke, ki jih lahko posreduje. Definiranih je šest tipov agentov.

- Pasivna naprava: je naprava, ki zgolj beleži in posreduje podatke. Primer takšnega agenta je senzor temperature zraka.
- Aktivna naprava je naprednejši agent in omogoča upravljanje z napravo. Primer takšnega agenta je toplotna črpalka, ki se jo lahko prižiga, ugaša in nastavlja na določeno temperaturo.
- Program, ki združuje agente, ki ponujajo različne storitve znotraj arhitekture. V to kategorijo spadajo agent dnevnik, ki beleži spremembe v sistemu, agent upravljanja sistema, agenti za učenje, agenti za optimizacijo, simulacijo ipd.
- »Hišniki« so upravljalški agenti, ki omogočajo nadzor in upravljanje določenih sklopov pametne stavbe. Odvisno od razdelitve in same hierarhične strukture, lahko ti agenti upravljajo sistem ogrevanja, določeno sobo ali pa celotno zgradbo.
- Podporni agenti, ki poskrbijo za pravilno konfiguracijo okolja in opravljajo generične funkcije, ki so na voljo vsem agentom.
- Komunikacijski agenti, ki skrbijo za prenašanje sporočil med agenti, beleženje posameznih agentov, posredovanje podatkov ipd.

Simbolična shema arhitekture je prikazana na sliki 3.11. Arhitektura je zasnovana modularno in omogoča enostavno dodajanje in odstranjevanje naprav v sistem oziroma iz njega.



Slika 3.11: Prikaz upravljanja z energijo in izmenjave z omrežjem.

Agentna arhitektura je implementirana na odprtokodni platformi Jade [24]. Jade je v celotni implementirana v programskem jeziku Java [25] in zahteva vsaj verzijo Jave 5. Jade poenostavlja izvajanje večagentnih sistemov in je skladen s specifikacijami FIPA (The Foundation for Intelligent Physical Agents) [26] in z naborom grafičnih orodij, ki podpirajo odpravljanje napak. Sistem, ki temelji na Jade, se lahko razporedi na več računalnikih. Konfiguracijo je moč nadzorovati s pomočjo oddaljenih grafičnih uporabniških vmesnikov. Poleg abstrakcije agenta Jade zagotavlja neposredno komunikacijo med agenti, ki temelji na paradigmi asinhronnega pošiljanja sporočil, ter številne druge napredne funkcije, ki omogočajo razvoj porazdeljenih agentnih sistemov [24].

Agent program je običajen javanski razred, ki razširja Jade razred Agent, kar povzroči asinhrono izvajanje in potrebo po implementaciji metode *action*.

Agent vremenska postaja pridobi vremensko napoved preko portala “Il Meteo” [27], ki obsega napovedi temperature, moči in smeri vetra, vlažnosti in sončnega obsevanja za okvirno teden dni naprej. Stran ne ponuja API-ja (Application Programable Interface), ki bi omogočal avtomatsko pridobivanje podatkov preko metod HTTP (The Hypertext Transfer Protocol), zato

smo podatke programsko pridobivali iz strukture HTML strani, za kar smo uporabili javansko knjižnico Jsoup [28].

Zaradi asinhrona narave agentov izvajanje pogosto nadziramo s postavljanjem zastavic. Osrednji agent optimizacije je optimizacijski agent, ki ima definiranih 10 stanj. V prvih nekaj stanjih optimizacijski agent od ostalih agentov pridobi potrebne podatke za začetek optimizacije: datum, vremensko napoved, podatke o omrežju, želje uporabnika in njegove nastavitve. Poleg pomembnih podatkov za potek optimizacije se v prvih nekaj stanjih definira komunikacija med agenti in agentu pridobi informacije za dostop do simulatorja stavbe. Akcija določenega obnašanja agenta (definirana v metodi *action*) se neprestano izvaja, zato, če želimo, da se vsa koda ne izvede, ob vsaki ponovitvi vpeljemo stanja. Stanja označujejo aktivni del kode. Metoda *action* sestoji iz enega velikega *switch* stavka. Vsako stanje je ena celoštevilska možna vrednost spremenljivke stanja, ki je sprva inicializirana na 1. Prikazan je generičen primer stanja, katerega strukturo si delijo vsa stanja.

@Override

```
public void action(){
    switch(state){
        case n: {
            if(!agent.flagSimulating){
                // ce ne poteka simulacija
                // izvedi kodo
                state++
            }
            break;
        }
    }
}
```

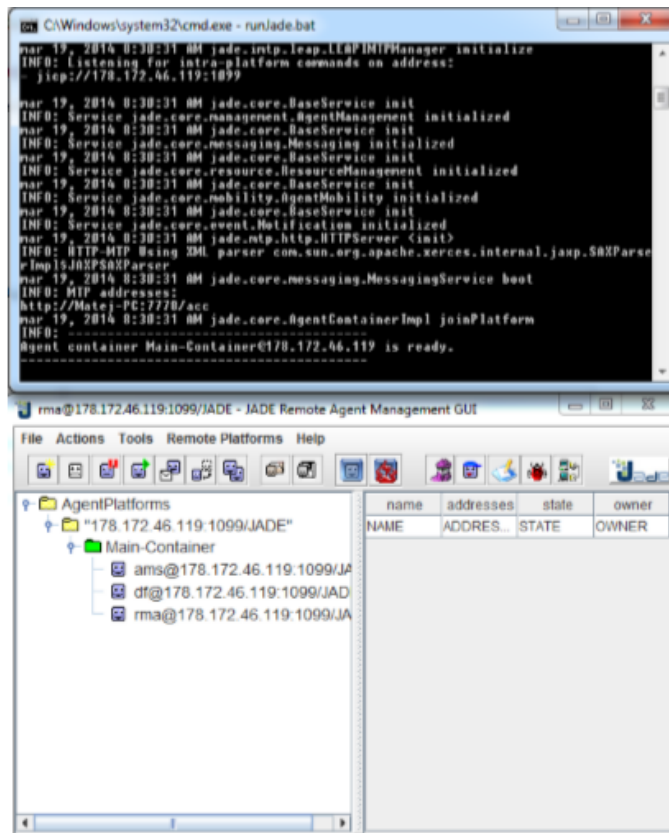
V našem primeru izvajanje simulacije urnika poteka nekaj sekund. Ker je za potek algoritma kritičnega pomena, da algoritem počaka na oceno novo

generiranega osebka, saj bo ta ocena vplivala na nadaljni potek, se ob začetku simulacije postavi zastavica *flagSimulating*, ki agentu sporoča, naj ne nadaljuje z izvajanjem algoritma. Čakanje v asinhronem agentnem okolju je impementirano s pogojnim stavkom izvajanja *if(!agent.flagSimulating)*. V primeru, da se izvaja simulacija, se izvede prekinitveni stavek *break*, kar sproži ponovno izvajanje metode *action*, vse dokler se simulacije ne zaključi in takrat izvedemo potrebno kodo. Stanje nato povečamo, da se bo naslednjič izvedlo stanje $n + 1$.

3.4 Postopek

Pametna stavba se mora dobro obnašati v veliko različnih pogojih, zato smo algoritem preizkusili na več testnih scenarijih. Poskuse smo opravili z sončno in pretežno oblačno vremensko napovedjo. Pretežna oblačna napoved pomeni, da je v prvem od obeh dni kljub oblačnosti čez dan kar nekaj sončnega obsevanja, drugi dan pa je veliko bolj oblačen. Za vsako od obeh možnih napovedi smo opravili poskuse, kjer je na začetku optimizacijskega okna akumulator skoraj poln, in scenarij, kjer je akumulator prazen. Poleg tega smo izvedli še poskus, kako se stavba obnaša, če pri upravljanju z energijo upošteva podatke o omrežju in temu primerno nastavi obnašanje pri vseh urnikih enako. Scenarij tako vsebuje: vremensko napoved (sončno/oblačno), stanje akumulatorja (poln/prazen) in stikalo, ki pove ali stavba uporabi podatke o omrežju (da/ne). Isti scenarij smo pognali večkrat z različnimi naključnimi semeni.

Pred zagonom optimizacije je potrebno vzpostaviti platformo, ki ustvari okolje, ki že vsebuje nekatere agente. Pojavi se uporabniški vmesnik, prikazan na sliki 3.12, ki prikazuje aktivne agente.



Slika 3.12: Grafični vmesnik Jade.

Pridobivanje podatkov za analizo poteka v več fazah.

- Med samim delovanjem algoritma program izpisuje za vsak simuliran urnik njegove stroške, udobje in ostanek energije v akumulatorju pretvorjen v potencialno vrednost. Poleg izpisovanja vseh generiranih rešitev se na datoteko *populacija.txt* shranjujejo vsi urniki, ki so na koncu trenutne iteracije v populaciji. Urniki se zapišejo v obliki *stroski,neudobje*.
- Ob vsaki simulaciji EnergyPlus ustvari datoteko csv, kjer se nahajajo informacije o dejanskem dogajanju v stavbi ob simulaciji. Za vse rešitve ni smiselno hraniti teh podatkov, zato se ob zaključku optimizacije nedominirane urnike ponovno simulira in shrani to datoteko, kar nam

daje dober vpogled v dogajanje v stavbi. Datoteka za vsak interval hrani vrednosti mnogo parametrov, med najpomembnejšimi so:

- neposredno sončno obsevanje na kvadratni meter sončnih panelov $[\text{W}/\text{m}^2]$,
- poraba električne energije stavbe $[\text{W}]$,
- poraba električne energije porabnika $[\text{W}]$,
- nastavljena temperatura $[\text{°C}]$,
- dejanska dosežena temperatura v stavbi $[\text{°C}]$,
- količina shranjene energije v akumulator $[\text{W}]$,
- količina izčrpane energije iz akumulatorja $[\text{W}]$,
- kumulativno stanje akumulatorja $[\text{W}]$,
- količina kupljene energije $[\text{W}]$,
- količina prodane energije $[\text{W}]$,
- udobje uporabnika.

Za vsak urnik poleg datoteke csv shranimo tudi dejanski urnik. S pomočjo teh dveh datotek je možna ustrezna analiza posameznega urnika.

- Hipervolumen je prostor, ki ga definirajo točke na Pareto optimalni fronti in izhodiščna točka. Za izhodiščno točko je bila uporabljena točka $I = (1.1, 1.1)$. Podatki iz datoteke *pupulacija.txt* se normalizirajo na interval $(0, 1.1]$. Normalizacijo izvedemo sami, za izračun hipervolumna je bila uporabljena odprtokodna implementacija v jeziku C Laboratorija za umetno inteligenco univerze v Bruslju [29].

Pridobivanje podatkov o hipervolumnu rešitev ni zaželeno v agentnem sistemu, ko bi se začel uporabljati v praksi, zato je pridobivanje podatkov za analizo možno s spremembo ene spremenljivke izključiti. Normalizacija podatkov za računanje hipervolumna je ločen program in ni del agentne arhitekture.

Poglavje 4

Analiza in vrednotenje rezultatov

S poganjanjem algoritma v opisanih scenarijih smo pridobili podatke, kako algoritem deluje z različnimi vhodnimi podatki. V poglavju 4.1 so predstavljeni rezultati in kratka analiza za vsak scenarij. V poglavju 4.2 primerjamo delovanje algoritma v različnih scenarijih in kritično ovrednotimo pridobljene rezultate.

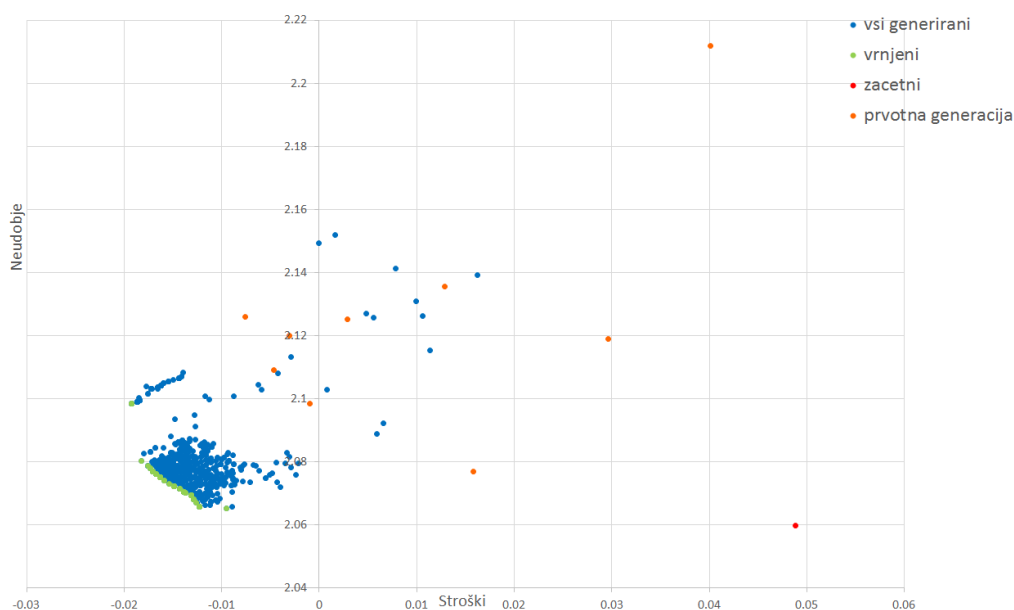
4.1 Analiza

Analizo rezultatov predstavimo v dveh delih. V razdelku 4.1.1 je predstavljena primerjava delovanja algoritma z uporabo podatkov o omrežju in brez uporabe tega znanja. V razdelku 4.1.2 primerjamo razlike med delovanjem algoritma glede na različna začetna stanja polnosti akumulatorja.

4.1.1 Znanje o omrežju

Primerjamo podatke o omrežju za določanje upravljanja z energijo. Zanima nas kako uspešno bo stavba upravljala z energijo, zato smo primerjavo izvedli z na začetku praznim akumulatorjem.

Vsak urnik je v prostoru rešitev predstavljen kot ena točka. Abscisna komponenta so stroški, ki jih urnik predstavlja, ordinatna komponenta pa neudobje, ki ga urnik povzroči uporabniku. Rdeča točka ustreza začetnemu urniku, ki ga preko uporabniškega vmesnika nastavi uporabnik. Oranžne točke predstavljajo urnike, ki so člani prvotne generacije. Zelene točke v grafu ustrezajo urnikom, ki ležijo na približku Pareto optimalne fronte. Ob koncu delovanja optimizacijskega algoritma se uporabniku ti urniki prikažejo preko uporabniškega vmesnika in uporabnik lahko za delovanje stavbe izbere poljubnega med njimi. Modre točke so ostali med optimizacijo generirani urniki. Na sliki 4.1 vidimo, da so stroški negativni, kar pomeni, da stavba z upravljanjem z energijo služi.

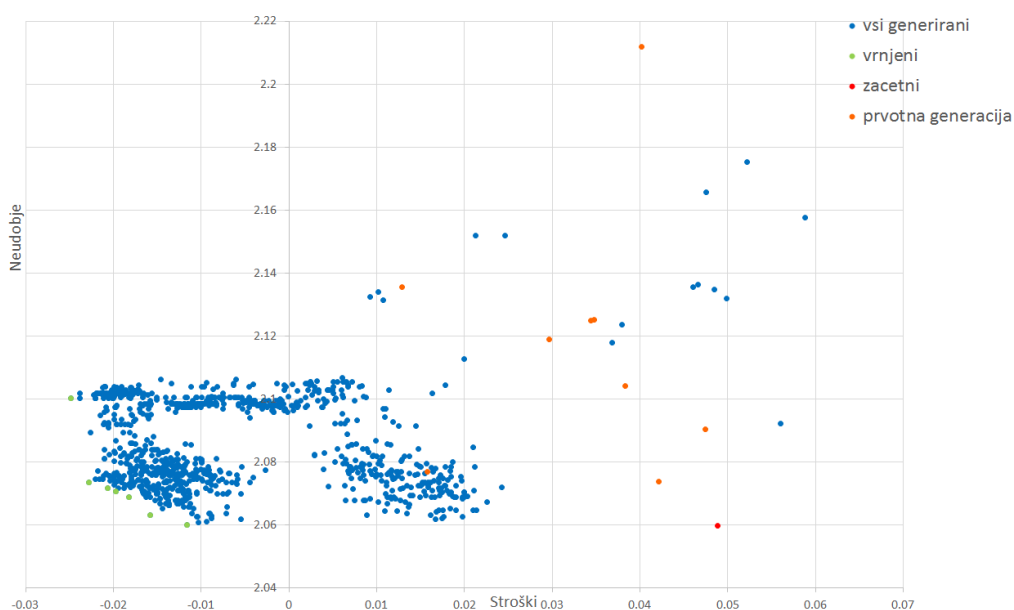


Slika 4.1: Prostor kriterijev ob sončnem vremenu z uporabo podatkov o omrežju.

Na koncu optimizacijskega okna lahko z izbranim urnikom pridobimo med 0,01 € in 0,02 €. Ob sončnem vremenu fotovoltaična panela proizvedeta zadostno količino energije, da jo stavba prodaja v omrežje.

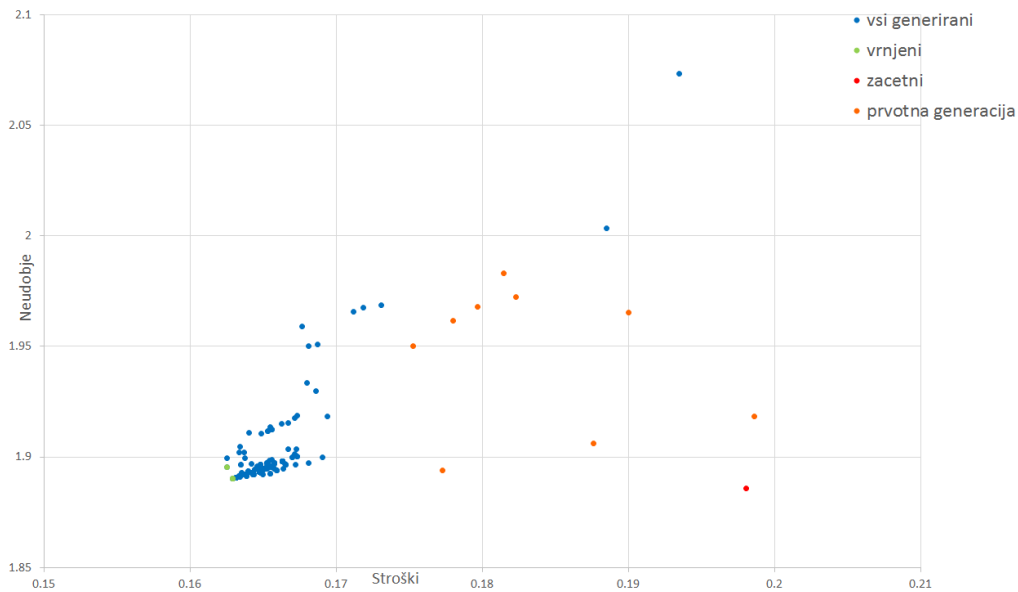
Na sliki 4.2, ki prikazuje optimizacijo brez uporabe podatkov o omrežju,

opazimo veliko razliko v primerjavi s sliko 4.1. Urniki so veliko bolj razpršeni, predvsem po abscisi. Razpršenost urnikov je pričakovana, saj imajo v primeru uporabe znanja o omrežju vsi urniki enake ukaze za upravljanje z energijo. Brez uporabe znanja potrebuje optimizacijski algoritem več generacij, da uspe najti strategijo upravljanja z energijo.



Slika 4.2: Prostor kriterijev ob sončnem vremenu brez uporabe podatkov o omrežju.

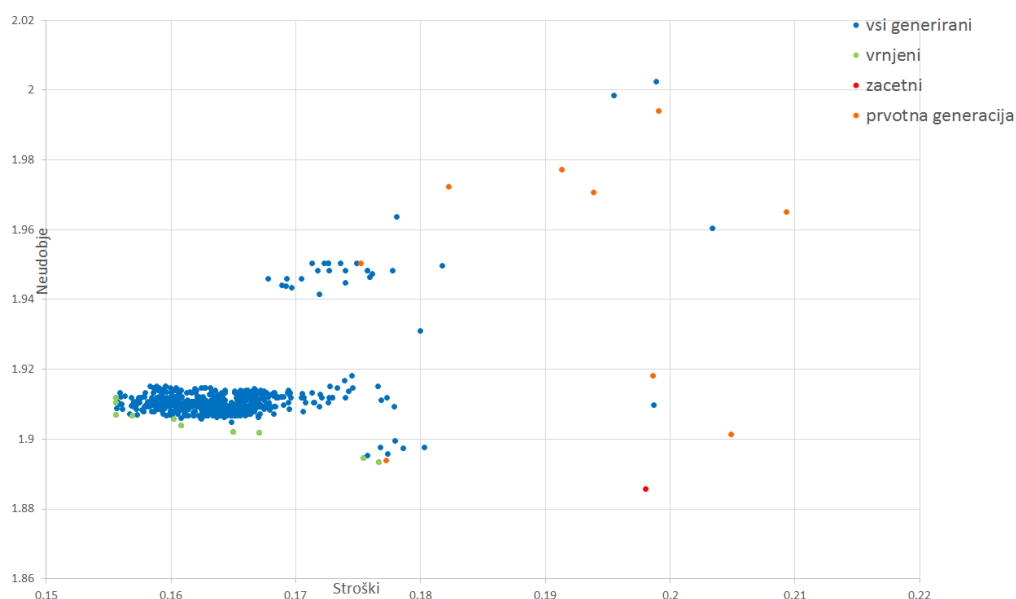
V primeru oblačnega vremena ni moč pričakovati, da bo stavba uspela delovati z dobičkom. Iz slike 4.3 je razvidno, da fotovoltaična panela ne proizvedeta zadostne količine energije za napajanje celotne stavbe oziroma za prodajanje le-te.



Slika 4.3: Prostor kriterijev ob oblačnem vremenu z uporabo podatkov o omrežju.

Stroški se gibljejo okoli 0,16 €. Na pogled je modro obarvanih urnikov, ki prikazujejo vmesne točke optimizacije, veliko manj kot pri obeh zgornjih primerih, kar je posledica velike podobnosti med ocenami urnikov. Razlog za to podobnost je, da imajo vsi enako strategijo upravljanja z energijo, na stroške oziroma udobje vplivajo samo spremembe glede porabnikov in temperature.

Rezultati simulacije pri oblačnem vremenu brez uporabe znanja o omrežju so prikazani na sliki 4.4. Opazimo raznovrstnost rešitev predvsem po stroškovni komponenti. Rezultati so skladni s poskusi pri sončnem vremenu brez uporabe znanja o omrežju, saj so veliko bolj raznoliki kot pri scenarijih z uporabo le-tega. Stroški Pareto optimalnih urnikov v tem scenariju znašajo med 0,16 € in 0,18 €.

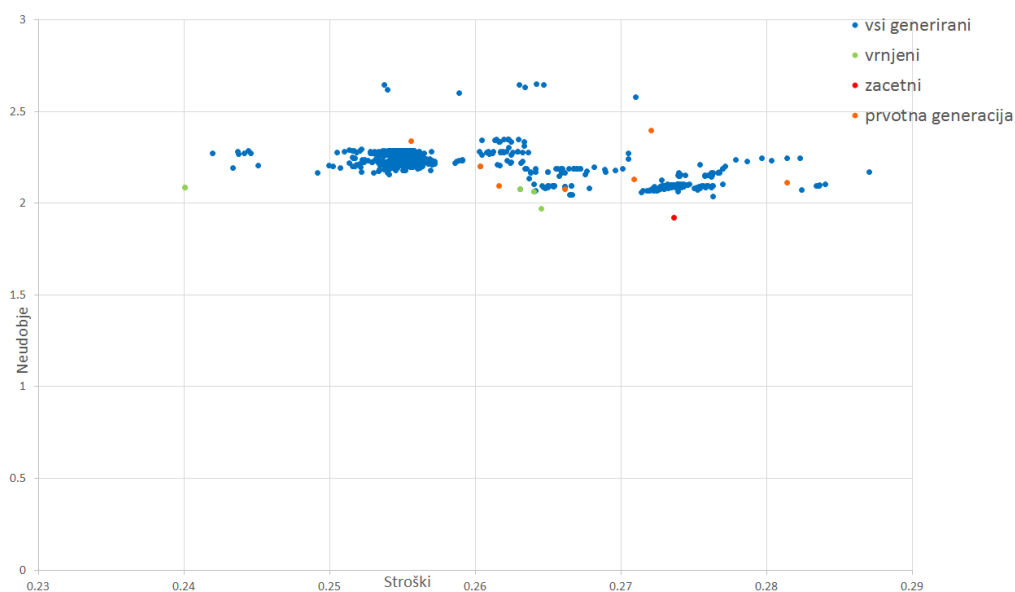


Slika 4.4: Prostor kriterijev ob oblačnem vremenu brez uporabe podatkov o omrežju.

4.1.2 Analiza vpliva začetnega stanja akumulatorja

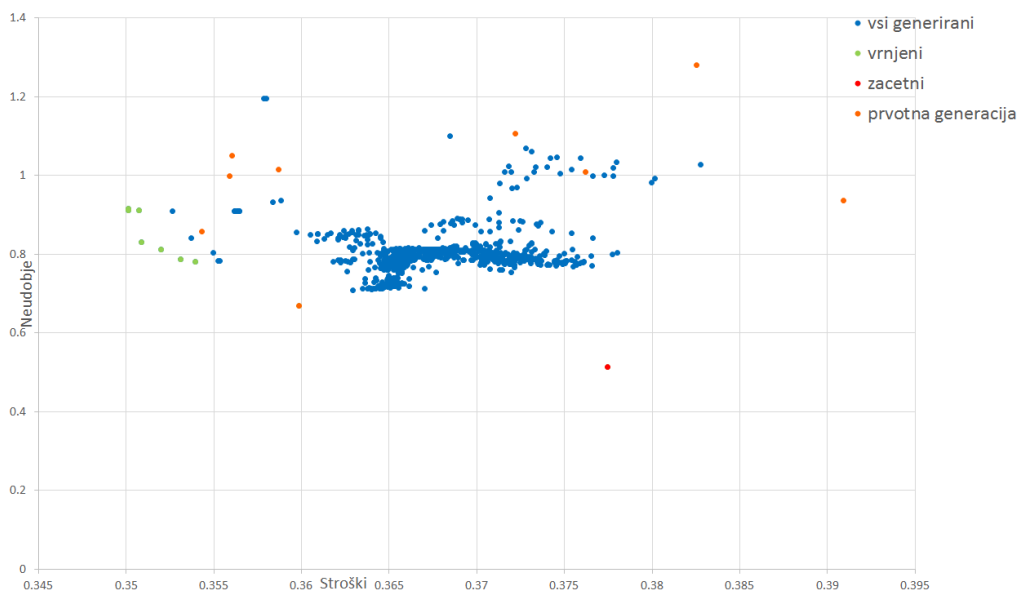
Namen naslednjih poskusov je primerjava obnašanja stavbe, z na začetku optimizacijskega okna, polnim in praznim akumulatorjem. Vsi poskusi so bili opravljeni z uporabo podatkov o omrežju. Za otežitev poskusov in povečanje števila konfiguracij smo povečali število porabnikov z dodatnimi omejitvami. V enem dnevu je potrebno zagnati pet porabnikov. Trikrat je različne porabnike potrebno pognati med 1:00 in 23:00, dvakrat pa med 17:00 in 20:00. Rezultate prikazujejo slike 4.5 do 4.10.

S povečanjem števila porabnikov v stavbi je energetska učinkovitost stavbe še bolj odvisna od pametnega razporejanja začetnega časa delovanja porabnikov. Posledica je vidna v veliko večjih spremembah stroškovne komponente urnikov v primerjavi z rezultati v 4.1.1. Točke na sliki 4.5 so močnejše razpršene v vodoravni smeri, saj je potrebnih več generacij, da algoritem najde optimalno strategijo razporejanja porabnikov tekom dneva.



Slika 4.5: Prostor kriterijev ob sončnem vremenu s praznim akumulatorjem.

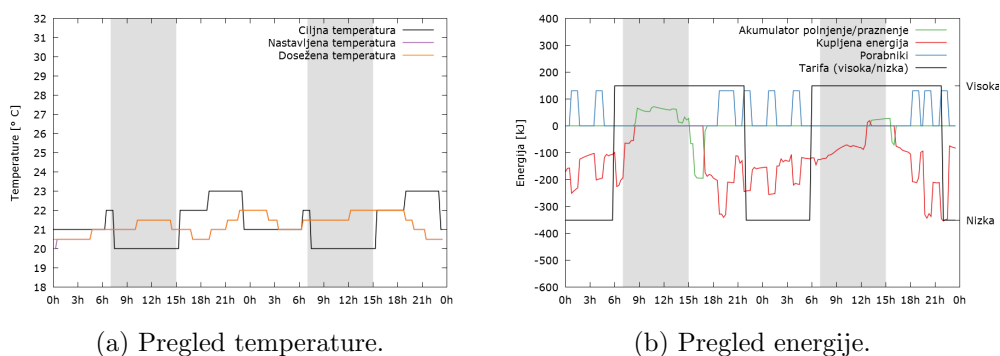
Na sliki 4.6 je, poleg primerjave s sončnim vremenom, dobro viden poziti-
tiven učinek načina generiranja začetne generacije.



Slika 4.6: Prostor kriterijev ob oblačnem vremenu s praznim akumulatorjem.

Pet urnikov je bilo v začetno generacijo dodanih kot rešitev prejšnjih optimizacij. Na preiskani prostor vplivata obe množici prvotne generacije, tako tista, ki je bila ustvarjena s perturbiranjem začetnega urnika, kot tudi tista, ki je bila dobljena s prejšnjimi optimizacijami. Večina preiskanih rešitev je bilo med tema, na sliki 4.6 jasno ločenima množicama. Sklepamo lahko, da so to rešitve, ki so bile dobljene s križanjem osebkov obeh delov začetne generacije.

Slika 4.7 prikazuje podroben vpogled v dogajanje v stavbi pri določenem urniku. Na sliki 4.7(a) je prikazano temperaturno dogajanje.

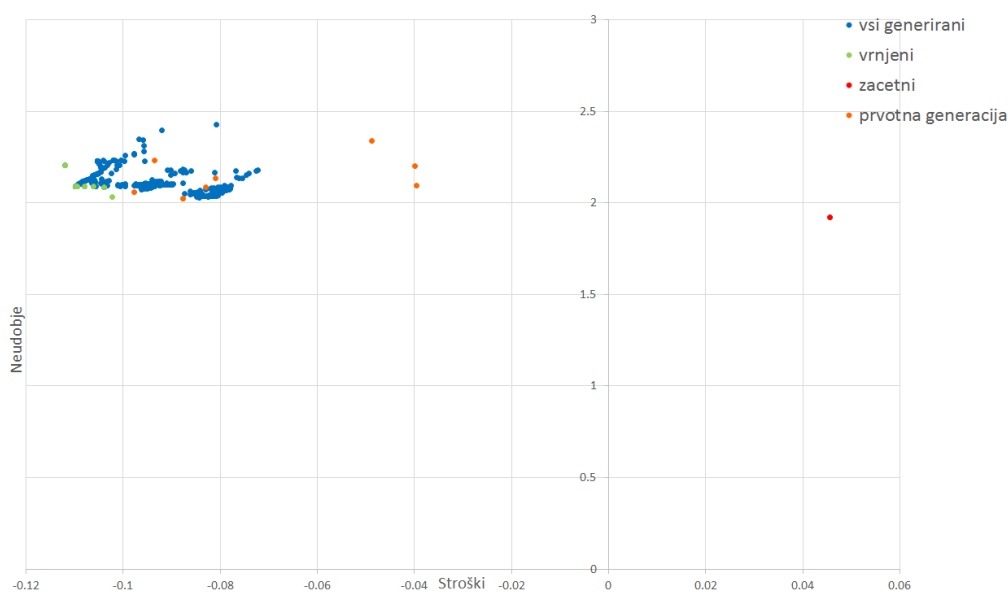


Slika 4.7: Podroben pregled najugodnejšega urnika.

Vijoličen graf predstavlja temperaturo, ki jo nastavi optimizacijski algoritem. V opisanem primeru se, z izjemo prvih nekaj intervalov, ta popolnoma ujema z doseženo temperaturo (oranžna barva), zato ni vidna. Črn graf predstavlja temperaturo, ki jo preko uporabniškega vmesnika nastavi uporabnik. Predeli z osenčenim ozadjem predstavljajo obdobja, ko uporabnik ni prisoten. Na sliki 4.7(b) je prikazano energetska delovanje. Moder graf prikazuje aktivnost porabnikov (neničelni intervali). Črn graf predstavlja obdobja nižje in višje tarife. Rdeč graf predstavlja količino odkupljene oziroma prodane energije v vsakem intervalu. Negativne vrednosti pomenijo črpanje energije iz omrežja, pozitivne vrednosti predstavljajo prodajanje energije. Opazi se ujemanje delovanje porabnikov z odvzemom energije (moder in rdeč graf). Zelen graf prikazuje dogajanje z akumulatorjem. Pozitivne vrednosti pome-

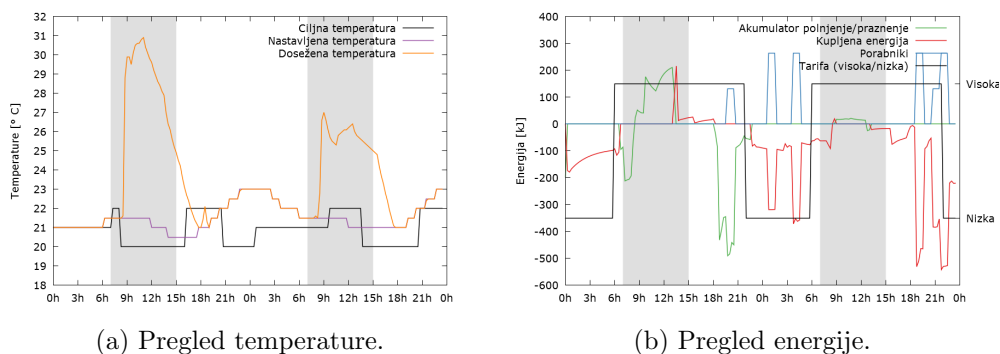
nijo, da se akumulator v tistem časovnem obdobju polni, negativne vrednosti predstavljajo črpanje energije iz akumulatorja.

Stroškovno najboljši scenarij bi moral biti prikazan na sliki 4.8, saj imamo akumulator na začetku optimizacijskega okna poln in sončno vremensko napoved.



Slika 4.8: Prostor kriterijev ob sončnem vremenu s polnim akumulatorjem.

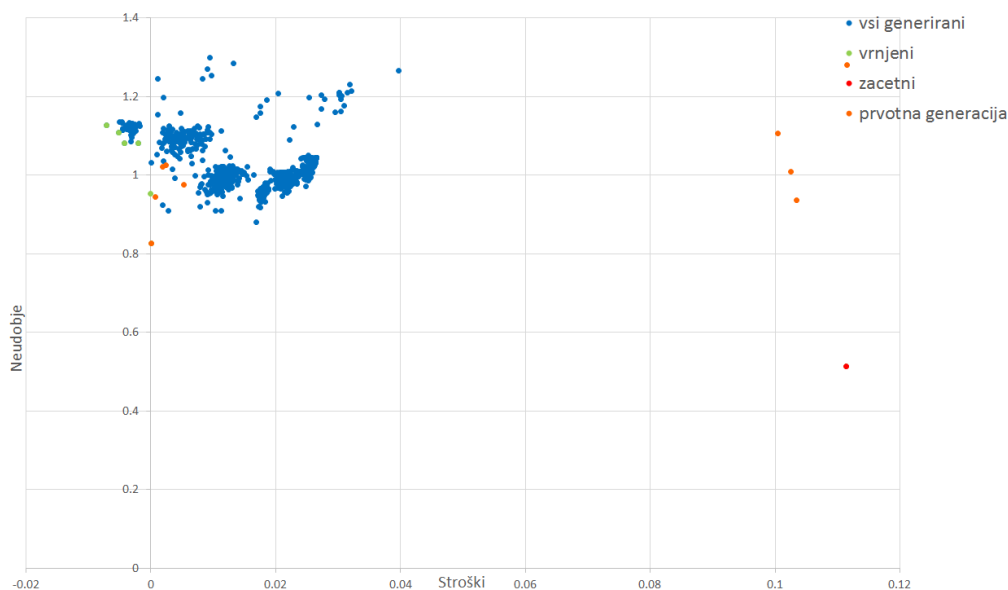
Izkaže se, da so kljub poganjanju petih porabnikov dnevno razmere dovolj ugodne, da stavba uspe delovati z dobičkom. Pareto optimalni urniki uspejo zaslužiti nekaj več kot 0,10 €. Prostor kriterijev je podoben tistemu na sliki 4.1, saj gre za isti scenarij, a z več porabniki. H končnim rešitvam pripomore dobro generirana začetna populacija. Na sliki 4.9 je podroben prikaz delovanja najugodnejšega urnika tega scenarija.



Slika 4.9: Podroben pregled najugodnejšega urnika.

Pri pregledu temperature na sliki 4.9(a) opazimo, da se v času odsotnosti uporabnika temperatura dvigne na 31 °C. Razlog je močno sončno obsevanje in posledično segrevanje prostora, ki ga ne ohlajamo, saj uporabnik ni prisoten in stavba ne sledi nastavljeni temperaturi. Hlajenje prostora med uporabnikovo odsotnostjo bi povzročilo višje stroške in nič več udobja. Prostor se začne ohlajati nekaj ur pred uporabnikovim prihodom, a zaradi učinka tople grede toplotna črpalka do uporabnikovega prihoda ne uspe stavbo ohladiti do želene temperature. Kar dela ta urnik še posebej zanimiv, je razporejanje porabnikov. Na pogled je pri pregledu energije na sliki 4.9(b) aktivnih manj uporabnikov od specifikacij, a v primerjavi s sliko 4.7(b) opazimo, da so vrhovi, ki kažejo aktivnost, višji. Algoritem je v uspel v čas nižje tarife razvstiti več porabnikov hkrati in s tem prihranil energijo.

Slika 4.10 prikazuje scenarij oblačnega vremena s polnim akumulatorjem. Viden je učinek polnega akumulatorja, saj kljub oblačnemu vremenu stavba še vedno služi z delovanjem.

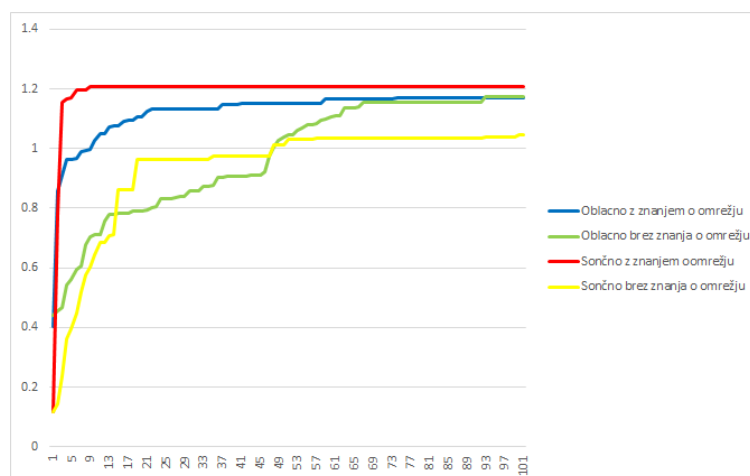


Slika 4.10: Prostor kriterijev ob oblačno vremenu s polnim akumulatorjem.

Sončnega obsevanja je dovolj, da stavba s črpanjem iz akumulatorja oziroma iz omrežja v času nižje tarife uspe ohraniti več energije v akumulatorju, kot jo je porabila. Prikazan dobiček v resnici ni dobiček, ampak gre za ostanek energije v akumulatorju, ki se odšteje od dejanskih stroškov delovanja stavbe.

4.2 Ovrednotenje rezultatov

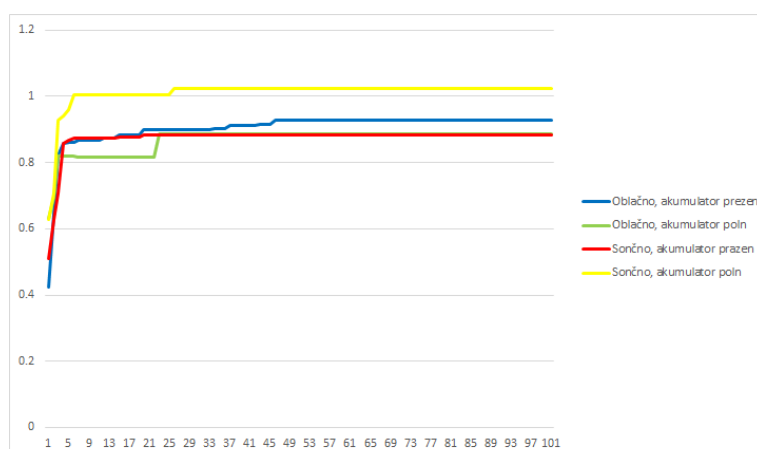
Iz rezultatov predstavljenih v razdelku 4.1 je razvidno, da optimizacijski algoritem v vseh prikazanih scenarijih uspe najti več kompromisnih rešitev med stroški in udobjem, ki so nedominirane. Primerjava rezultatov iz razdelka 4.1.1 je prikazana z doseženimi hipervolumni rešitev na sliki 4.11. Opazne so razlike med oblačnim in sončnim vremenom ter med uporabo znanja o omrežju oziroma njegovo neuporabo.



Slika 4.11: Primerjava hipervolumnov različnih scenarijev iz 4.1.1.

Na sliki 4.11 vidimo podobnost med rdečim in modrim grafom. Oba predstavljata poskus z uporabo znanja o omrežju in narasteta veliko hitreje kot ostala dva. V scenarijih brez uporabe znanja o omrežju je napredovanje generacij veliko počasnejše. Kljub počasnejši konvergenci v primeru oblačnega vremena uspeša oba pristopa priti do podobnih rešitev. Graf scenarija sončnega vremena brez znanja o omrežju kaže, da je strategija prodajanja energije za optimizacijski algoritem problematična. Algoritmu ne uspe najti ustreznih rešitev, ki bi v zadostni meri sledile strategiji, ki je opisana v razdelku 3.2.2.

Na sliki 4.12 so prikazani doseženi hipervolumni pri poskusih iz razdelka 4.1.2. Opazimo, da so si rezultati bolj podobni kot v razdelku 4.1.1. Algoritem uspe konsistentneje oziroma hitreje konvergirati k določeni množici rešitev. Začetni in končni urniki so med sabo bolj podobni, kar se kaže v nižjem doseženem hipervolumnu. Glavni razlog je uporaba znanja o omrežju. Vidna je podobnost dveh grafov z uporabo znanja o omrežju s slike 4.11 in z vsemi s slike 4.12.



Slika 4.12: Primerjava hipervolumnov scenarijev iz 4.1.2.

Optimizacijski algoritem na račun udobja najde mnogo stroškovno ugodnejših rešitev. Stroškovna primerjava glede na začetni urnik v vsakem od scenarijev v razdelku 4.1.2 je prikazana v tabeli 4.1.

Tabela 4.1: Stroškovna razlika med začetnim in najugodnejšim urnikom.

	sončno vreme	oblačno vreme
akumulator prazen	0,034 €	0,027 €
akumulator poln	0,158 €	0,118 €

Opazimo, da algoritem uspe s polnim akumulatorjem na začetku optimizacijskega okna najti, v primerjavi s praznim akumulatorjem, veliko ugodnejši urnik od začetnega. Na prvi pogled so optimizirani zneski nizki. Upoštevati je potrebno dejstvo, da je prostor velik le $12,45 \text{ m}^2$. Z velikostjo prostora se kompleksnost rešitve večja in verjetno ni pravilno predvidevati linearnega modela stroškovne učinkovitosti. Poleg velikosti prostora je treba upoštevati še število porabnikov ter njihovo zmogljivost, kakor tudi uporabljene materiale in njihove izolacijske lastnosti. Kljub temu je v tabeli 4.2 za predstavo in umestitev pridobljenih rezultatov v perspektivo uporabljen linearni model odvisnosti med velikostjo prostora in stroškovno optimizacijo.

Tabela 4.2: Linearno predvidevanje stroškovnih optimizacij.

	14,5 m ²	80 m ²	160 m ²
1 mesec	1,27 €	7,01 €	14,02 €
1 leto	15,24 €	84,12 €	168,24 €

Za izračun stroškov v enem mesecu je bil vzet povprečen mesec z dvajsetimi sončnimi in desetimi oblačnimi dnevi. Letni izračun predstavlja dvanajst opisanih mesecev. Kljub navidez majhnim prihrankom s projekcijo rezultatov na večjo stavbo vidimo, da niso zanemarljivi, saj 84 € v 80 m² velikem stanovanju na letni ravni ni malo.

Kljub različnim scenarijem so vrednosti neudobja razmeroma konstantne. Primerjava neudobja urnikov je prikazana v tabeli 4.3. Neudobje je vsota temperaturnih odmikov dejanske temperature od uporabikove želene temperature preko vseh intervalov.

Tabela 4.3: Primerjava najudobnejšega in najugodnejšega urnika po neudobju.

	najugodnejši	najmanj udoben	razlika
sončno, akumulator prazen	1,922	2,206	0,284
sončno, akumulator poln	1,922	2,087	0,165
oblačno, akumulator prazen	0.513	1.127	0.614
oblačno, akumulator poln	0.513	0.912	0,399

Vrednosti neudobja v sončnem vremenu so okoli dvakrat višje od vrednosti neudobja pri oblačnem vremenu. Večino tega lahko pripišemo dejstvu, da stavba ob neprisotnosti uporabnika ne uspe ohladiti prostora dovolj zgodaj, da bi bila ob prihodu ta že idealna (videno pri pregledu posameznega urnika ob sončnem vremenu na sliki 4.7(a)). V primeru oblačnega dneva so temperaturne razlike nižje in povzročijo manj neudobja. Do velikih razlik pride zaradi izračuna neudobja, ki je razlika med dejansko temperaturo in želeno. Treba je upoštevati, da je prikazan najslabši urnik z vidika udobja (ima

maksimalno neudobje), ki je žrtvoval največ udobja na račun minimizacije stroškov.

Poglavje 5

Zaključek

Cilj diplomskega dela je bila optimizacija delovanja pametne stavbe. Tekom projekta smo uspešno razvili algoritem, ki je sposoben optimizirati delovanje pametne stavbe in upošteva kriterija uporabnikovega udobja in stroškov delovanja. Za vrednotenje ustvarjenih urnikov delovanja stavbe je bil uporabljen simulator, ki je upošteval model definirane stavbe, vremensko napoved, ceno električne energije, uporabnikove navade in želje. Predstavili smo razvit večkriterijski evolucijski algoritem, ki uporablja splošne in problemu prilagojene genetske operatorje. Razvita je bila strategija pametnega upravljanja z električno energijo, ki se je izkazala za cenovno ugodno.

Algoritem je bil preiskujen s številnimi vhodnimi podatki. Poskuse smo opravili v sončnem in oblačnem vremenu, opravljena je bila študija vpliva polnosti akumulatorja na začetku optimizacijskega okna in uporabe razvite strategije pametnega upravljanja z energijo. V vsakem testnem scenariju je algoritem uspel najti več kompromisnih rešitev, ki so bile Pareto optimalne. Začetnemu urniku, ki ga je vnesel uporabnik in je zato najudobnejši, je algoritem uspel najti rešitve, ki so bile manj udobne, a cenovno ugodnejše. Prikazali smo, da so na pogled majhni cenovni prihranki posledica majhnosti modela stavbe. Ob projekciji na večje stavbe, bi lahko prihranili tudi več kot 100 € letno.

Za nadaljne delo bi bilo smiselno testne scenarije še otežiti in jih čimbolj

približati realnosti. Preveriti bi bilo treba delovanje v nepredvidljivih in spremenljivih okoljih. V uporabljenem simulatorju ni bilo moč implementirati nepredvidljivih dogodkov, kot na primer, da uporabnik pride domov prej ali pa se vreme ne sklada z napovedjo. Smiselna bi bila sprememba merjenja uporabnikovega udobja. Namesto zgolj matematičnega odmika dejanske temperature od želene bi bilo bolje uporabiti modele uporabnikovega udobja, ki poleg notranje temperature upoštevajo še relativno in absolutno vlažnost zraka, hitrost zunanjega vetra, letni čas, zunanjo temperaturo in še druge podatke.

Projekt je bil namenjen tudi študiji primernosti uporabe evolucijskih algoritmov za optimiziranje delovanja pametne stavbe. Menimo, da smo z rezultati delovanja primernost prikazali na jasen, kritičen in razumljiv način. Razvita rešitev pa kljub prikazani uspešnosti še ni pripravljena za integracijo v pametno stavbo. Potrebno je še veliko razvoja na področju napovedovanja uporabnikovih navad in obvladovanja nepredvidljivih situacij v realnem svetu. Največja težava dosedanjega dela je v tem, da je celotno dogajanje razmeroma izolirano in predvidljivo, kar lahko vodi v rešitve, ki izgledajo odlične, a popolnoma odpovedo v primeru sprememb v okolju. Naša želja je, da bo algoritem v prihodnosti zmožen upoštevati tudi spremembe v okolju in bo to vodilo v robustno delovanje stavbe. Verjetno v prihodnosti večkriterijski algoritem za generiranje urnikov za daljše obdobje ne bo samostojna rešitev, ki bi upravljala s stavbo, ampak se bodo algoritmi povezovali s hitro razvijajočim se konceptom interneta stvari (angl. Internet of Things), ki mnogo obeta na področju hišne avtomatizacije.

Literatura

- [1] R. Slovenija, “Zakon o ratifikaciji Kjotskega protokola k Okvirni konvenciji Združenih narodov o spremembi podnebja (17/2002),” 2002.
- [2] J. Verbič, T. Poje “Vpliv kmetijstva na kakovost zraka in podnebne spremembe.” Predstavitev, 2006.
- [3] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008.
- [4] *Annual Energy Outlook*, US Energy Information Administration, June, 2012.
- [5] B. Rezaie and M. A. Rosen, “District heating and cooling: Review of technology and potential enhancements,” *Applied Energy*, vol. 93, pp. 2–10, 2012.
- [6] K. Fong, V. Hanby, and T. Chow, “System optimization for HVAC energy management using the robust evolutionary algorithm,” *Applied Thermal Engineering*, vol. 29, no. 11–12, pp. 2327–2334, 2009.
- [7] M. Söylemez, “On the optimum channel sizing for HVAC systems,” *Energy Conversion and Management*, vol. 42, no. 7, pp. 791–798, 2001.
- [8] S. Wang and Z. Ma, “Supervisory and optimal control of building HVAC systems: A review,” *HVAC&R Research*, vol. 14, no. 1, pp. 3–32, 2008.

- [9] K. F. Fong, V. I. Hanby, and T.-T. Chow, "HVAC system optimization for energy management by evolutionary programming," *Energy and Buildings*, vol. 38, no. 3, pp. 220–231, 2006.
- [10] J. Teeter and M.-Y. Chow, "Application of functional link neural network to HVAC thermal dynamic system identification," *Industrial Electronics, IEEE Transactions on*, vol. 45, no. 1, pp. 170–176, 1998.
- [11] J. A. Wright, H. A. Loosemore, and R. Farmani, "Optimization of building thermal design and control by multi-criterion genetic algorithm," *Energy and Buildings*, vol. 34, no. 9, pp. 959–972, 2002.
- [12] L. G. Caldas and L. K. Norford, "Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems," *Journal of Solar Energy Engineering*, vol. 125, no. 3, pp. 343–351, 2003.
- [13] J. Holland, *Adaption in Natural and Artificial Systems*, Ann Arbor, MI: The University of Michigan Press, 1975.
- [14] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [15] "Department of Intelligent Systems – Jožef Stefan Institute." <http://dis.ijs.si/>. [Dostopano: 26.08.2015].
- [16] "Projekt OpUS." <http://dis.ijs.si/Opus/>. [Dostopano: 26.08.2015].
- [17] G. Anogianakis, C. Bühler, and M. Soede, *Advancement of Assistive Technology*, IOS Press, 1997.
- [18] T. Šef, R. Piltaver, and T. Tušar, "Projekt OpUS: Optimizacija upravljanja energetske učinkovitih pametnih stavb," *Zbornik 16. mednarodne multikonference Informacijska družba*, pp. 110–113, 2013.
- [19] M. R. Kulkarni and F. Hong, "Energy optimal control of a residential space-conditioning system based on sensible heat transfer modeling," *Building and Environment*, vol. 39, no. 1, pp. 31–38, 2004.

-
- [20] R. Yang and L. Wang, “Multi-objective optimization for decision-making of energy and comfort management in building automation and control,” *Sustainable Cities and Society*, vol. 2, no. 1, pp. 1–7, 2012.
- [21] “Random (Java Platform SE 7).” <http://docs.oracle.com/javase/7/docs/api/java/util/Random.html>. [Dostopano: 16.08.2015].
- [22] “Building Technologies Office: EnergyPlus Energy Simulation Software.” <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. [Dostopano: 14.08.2015].
- [23] “Multi-agent system - Wikipedia.” https://en.wikipedia.org/wiki/Multi-agent_system. [Dostopano: 16.08.2015].
- [24] “Java agent development framework.” <http://jade.tilab.com>. [Dostopano: 14.08.2015].
- [25] “Learn why java is the language of choice for developers.” <https://java.com/en/about/>. [Dostopano: 21.08.2015].
- [26] “Welcome to the foundation for intelligent physical agents.” <http://www.fipa.org>. [Dostopano: 14.08.2015].
- [27] “Il Meteo.” <http://www.ilmeteo.it>. [Dostopano: 14.08.2015].
- [28] “Jsoup: Java HTML parser.” <http://jsoup.org/>. [Dostopano: 14.08.2015].
- [29] “Computation of the Hypervolume Indicator.” <http://iridia.ulb.ac.be/~manuel/hypervolume>. [Dostopano: 16.08.2015].